

*Приложение 2.31.1 к ОПОП по специальности
09.02.13 Интеграция решений с применением
технологий искусственного интеллекта*

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

Многопрофильный колледж

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ
УЧЕБНОЙ ДИСЦИПЛИНЫ
ОП.09 ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ**

**для обучающихся специальности
09.02.13 Интеграция решений с применением технологий искусственного интеллекта**

Магнитогорск, 2025

ОДОБРЕНО

Предметно-цикловой комиссией
«Информатики и вычислительной техники»
Председатель Т.Б. Ремез
Протокол № 5 от «22» января 2025г

Методической комиссией МпК

Протокол № 3 от «19» февраля 2025г

Разработчик (и):

реподаватель отделения №2 «Информационных технологий и транспорта»

Р.Н. Урманова

преподаватель отделения №2 «Информационных технологий и транспорта»

С.М. Утралинова

Методические указания по выполнению лабораторных работ разработаны на основе рабочей программы учебной дисциплины / профессионального модуля «ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ».

Содержание практических занятий ориентировано на подготовку обучающихся к освоению профессионального(ых) модуля(ей) программы подготовки специалистов среднего звена по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта (квалификация Специалист по работе с искусственным интеллектом) и овладению профессиональными компетенциями.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

4

2.МЕТОДИЧЕСКИЕ УКАЗАНИЯ.....6

Практическое занятие № 1 Анализ предметной области различными методами: контент- анализ, вебометрический анализ, анализ ситуаций, моделирование и др.	6
Практическое занятие № 2 «Изучение устройств автоматизированного сбора информации»	14
Практическое занятие № 3 «Оценка экономической эффективности информационной системы»	16
Практическое занятие № 4 Разработка модели архитектуры информационной системы	25
Практическое занятие № 5 «Обоснование выбора средств проектирования информационной системы»	32
Практическое занятие № 6 «Описание бизнес-процессов заданной предметной области»	36
Практическое занятие № 7 Построение модели управления качеством процесса изучения модуля Проектирование и разработка информационных систем»	42
Практическое занятие № 8 2 Реинжиниринг методом интеграции	44
Практическое занятие № 9 Разработка требований безопасности информационной системы	56
Практическое занятие № 10 Реинжиниринг бизнес-процессов методом горизонтального и/или вертикального сжатия	58
Практическое занятие № 11 Проектирование спецификации информационной системы индивидуальному заданию	62
Практическое занятие № 12 Разработка общего функционального описания программного средства по индивидуальному заданию	64
Практическое занятие № 13 Разработка руководства по инсталляции программного средства по индивидуальному заданию	66
Практическое занятие № 14 Разработка руководства пользователя программного средства по индивидуальному заданию	83
Практическое занятие № 15 Изучение средств автоматизированного документирования	98

1 ВВЕДЕНИЕ

Важную часть теоретической и профессиональной практической подготовки обучающихся составляют практические занятия.

Состав и содержание практических занятий направлены на реализацию Федерального государственного образовательного стандарта среднего профессионального образования.

Ведущей дидактической целью практических занятий является формирование профессиональных практических умений (умений выполнять определенные действия, операции, необходимые в последующем в профессиональной деятельности).

Ведущей дидактической целью лабораторных занятий является экспериментальное подтверждение и проверка существенных теоретических положений (законов, зависимостей).

В соответствии с рабочей программой ОП.09 ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ.

В результате их выполнения, обучающийся должен:

уметь:

У2. проводить анализ предметной области;

У3. осуществлять выбор модели и средства построения информационной системы и программных средств;

У4. использовать алгоритмы обработки информации для различных приложений; У9. работать с инструментальными средствами обработки информации;

Содержание практических и лабораторных занятий ориентировано на подготовку обучающихся к освоению профессионального модуля программы подготовки специалистов среднего звена по специальности и овладению **профессиональными компетенциями:**

ПК 2.2 Осуществлять процедуры администрирования баз данных

ОК 01 - Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

Выполнение обучающимися практических занятий по ОП.09 ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ.

направлено на:

- обобщение, систематизацию, углубление, закрепление, развитие и детализацию полученных теоретических знаний по конкретным темам учебной дисциплины;
- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;
- развитие интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;
- выработку при решении поставленных задач профессионально значимых качеств, таких как самостоятельность, ответственность, точность, творческая инициатива.

Практические и лабораторные занятия проводятся в рамках соответствующей темы, после освоения дидактических единиц, которые обеспечивают наличие знаний, необходимых для ее выполнения.

2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Раздел 1. Основы проектирования информационных систем Практическое занятие № 1

Анализ предметной области различными методами: контент-анализ, вебметрический анализ, анализ ситуаций, моделирование и др.

Цель: ознакомиться с процессом анализа предметной области и получить навыки по использованию методов анализа предметной области. Студенту предоставляется предметная область согласно выбранному варианту.

Выполнив работу, Вы будете:

уметь:

- У1. осуществлять постановку задач по обработке информации;
- У2. проводить анализ предметной области.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства.

Задание:

Произвести контент-анализ, вебметрический анализ, анализ ситуаций и моделирование предметной области, согласно выбранному варианту.

Порядок выполнения работы:

Список предметных областей:

1. Гостиница. Гостиница предоставляет номера клиентам на определенный срок. Каждый номер характеризуется вместимостью, комфортностью (люкс, полулюкс, обычный) и ценой. Вашими клиентами являются различные лица, о которых вы собираете определенную информацию (фамилия, имя, отчество и некоторый комментарий). Сдача номера клиенту производится при наличии свободных мест в номерах, подходящих клиенту по указанным выше параметрам. При поселении фиксируется дата поселения. При выезде из гостиницы для каждого места запоминается дата освобождения.
2. Агентство недвижимости. Агентство недвижимости занимается покупкой, продажей, сдачей в аренду объектов недвижимости по договорам с их собственниками. Агентство управляет объектами недвижимости как физических, так и юридических лиц. Собственник может иметь несколько объектов. В случае покупки или аренды клиент может произвести осмотр объекта. В качестве одной из услуг, предлагаемых агентством, является проведение инспектирования текущего состояния объекта для адекватного определения его рыночной цены. По результатам своей деятельности агентство производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
3. Кадровое агентство. Кадровое агентство способствует трудоустройству безработных граждан. Агентство ведет учет и классификацию данных о безработных на основании резюме от них. От предприятий города поступают данные о свободных вакансиях, на основании которых агентство предлагает различные варианты трудоустройства соискателям. В случае положительного исхода поиска вакансии считается заполненной, а безработный становится трудоустроенным. По результатам своей деятельности кадровое агентство производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
4. Компания по разработке программных продуктов. Компания заключает договор с

клиентом на разработку программного продукта согласно техническому заданию. После утверждения технического задания определяется состав и объем работ, составляется предварительная смета. На каждый проект назначается ответственный за его выполнение – куратор проекта, который распределяет нагрузку между программистами и следит за выполнением технического задания. Когда программный продукт готов, то его внедряют, производят обучение клиента и осуществляют дальнейшее сопровождение. По результатам своей деятельности компания производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики

5. Туроператор. Туроператор предоставляет возможность своим клиентам осуществить туристическую или деловую поездку в различные города России и мира. При разработке нового тура сначала анализируется текущая ситуация на рынке туризма и выбирается направление тура. После этого определяется статус тура, бронируются места в гостиницах и билеты на переезд к месту тура, разрабатывается культурная/деловая/развлекательная программа, утверждаются сроки тура. На каждый тур назначается ответственное лицо от туроператора, которое будет вести данный тур для улаживания проблем в случае возникновения каких-нибудь чрезвычайных или форс- мажорных ситуаций. Клиент приходит в офис туроператора, где вместе с менеджером выбирает уже разработанный тур и оформляет путевку. После возвращения из тура клиент может высказать свои замечания или пожелания, которые будут учтены при доработке существующих туров или при разработке

новых. Также, для дальнейшего улучшения тура, туроператор проводит анализ отчетов от посредников (гостиница, гиды и т.д.). По результатам своей деятельности туроператор производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.

1. Проанализировать предметную область, уточнив и дополнив ее, руководствуясь собственным опытом, консультациями и любыми источниками (книгами, учебниками или интернет- источниками).

2. После анализа необходимо выполнить структурное разбиение предметной области на подсистемы. Для каждой подсистемы определить задачи и функции. Указать входную и выходную информацию для каждой подсистемы.

3. Произвести контент-анализ предметной области, согласно выбранному варианту.

4. Произвести вебометрический анализ предметной области, согласно выбранному варианту.

5. Произвести анализ ситуаций предметной области, согласно выбранному варианту.

Произвести моделирование предметной области, согласно выбранному варианту.

Ход работы:

Структурный анализ

В основе структурного анализа лежит выявление структуры как относительно устойчивой совокупности отношений, признание методологического примата отношений над элементами в системе, частичное отвлечение от развития объектов.

Структурный элемент является основным элементом структурного анализа. Структурный элемент по-другому называют структурным объектом. Структурный элемент предназначен для выполнения одного из элементарных функций, связанных с моделируемым предметом, процессом или явлением.

Структурный анализ предполагает исследование предметной области с помощью графического представления модели системы. Данная система рассматривается как один большой объект, который в дальнейшем детализируется, приобретая иерархическую структуру.

Для такого подхода характерны:

- разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 9);
- указание сущности каждого уровня системы;

- использование строгих формальных правил записи;
- последовательное приближение к конечному результату.

Цель структурного анализа – это переход от абстрактных, общих знаний об исследуемой предметной области в точную модель.

Процесс декомпозиции является условным приемом, позволяющим представить систему в виде, удобном для восприятия, и оценить ее сложность. В результате декомпозиции подсистемы по определенным признакам выделяются отдельные структурные элементы и связи между ними. Процесс декомпозиции представляет собой разбиение одной большой системы на более мелкие подсистемы. Например, существует система «Колледж». В процессе декомпозиции можно выделить ряд подсистем, такие как отделения, рисунок 1.



Рисунок 1 – декомпозиция системы «Колледж»

Контент-анализ

Контент-анализ – метод количественного анализа содержания исследуемых текстов, применяемый для измерения в них каких-либо количественно определяемых характеристик.

В информационно-аналитической деятельности измеряются: частота ключевых слов и словосочетаний, служащих для описания объекта; повторяемость избранных объектов (имен, наименований фирм, марок продукции и др.) в некотором наборе документов/сайтов, объемы текстов; сила связи между публикациями по признакам социотирования; оценивается вклад научных коллективов в разработку объекта.

База исследования может включать вторичные или первичные документы разного целевого назначения: материалы конференций, обзорные публикации об объекте через какие-либо промежутки времени, новостные сообщения СМИ, поток правовых актов и комментариев к ним, рекламных материалов, авторефератов диссертаций.

Отбор единиц анализа основывается на семантических критериях. **Категории анализа** – это обобщающие ключевые слова и словосочетания, имеющиеся в тексте и соответствующие теме исследования.

В качестве единиц анализа принято считать: термины, понятия, ключевые слова; выраженные словами или словосочетаниями; темы, проблемы, тенденции развития; имена людей, названий организаций, географические названия, имена собственные.

Единица счета представляет собой количественную меру единицы анализа, которая регистрирует частоту появления признака категории анализа в тексте. Например, частота встречаемости единиц анализа в текстах, на сайтах, объем текстов, объем статьи, длительность трансляции по радио или ТВ; размер аудио- и видеозаписей и др.

Этапы контент-анализа:

1. Подготовительный этап. Включает определение цели, задач, гипотезы исследования; отбор источников информации и массива текстов для анализа, выбор категорий и единиц анализа, единиц счета, технологии обработки данных, разработка инструментов – форматов таблиц, систем кодирования, условных сокращений, инструкций аналитику и др. В числе обязательных инструментов рекомендуются:

- классификатор контент-анализа – общая таблица, в которую сведены все категории (подкатегории) анализа и единицы анализа; ее основное назначение – предельно четко

зафиксировать те единицы, в которых выражается каждая категория, используемая в исследовании;

- протокол (бланк) контент-анализа – содержит сведения о документе (его авторе, времени издания, объеме и т. п.); итоги его анализа (количество случаев употребления в нем определенных единиц анализа и следующие отсюда выводы относительно категорий анализа); протоколы заполняются, как правило, в закодированном виде;
- регистрационная карточка представляет собой кодировальную матрицу, в которой отмечается количество единиц счета, характеризующее единицы анализа; протокол контент-анализа каждого конкретного документа заполняется на основе подсчета данных всех регистрационных карт, относящихся к этому документу;
- инструкция аналитику (кодировщику) – система правил и пояснений для исполнителей по сбору эмпирической информации, регистрации заданных единиц анализа.

2. Сбор и первичная обработка данных контент-анализа. Производится процедура подсчета семантических единиц разного уровня обобщения. Фиксация данных при контент-анализе обычно осуществляется с помощью достаточно простых анкет, в которых фиксируется каждое появление в анализируемом тексте искомой единицы. Составляются счетные таблицы по отдельным текстам, а также сводные таблицы по отдельным категориям (темам).

Самый простой вариант контент-анализа – количественный подсчет частоты встречаемости ключевых слов в тексте (по массиву в целом, по отдельным группам публикаций, по отдельным единицам наблюдения и др.). Также может быть определен удельный вес той или иной категории (К), который можно вычислить с помощью формулы 1:



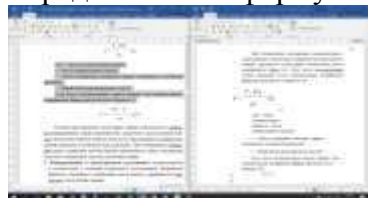
Для установления соотношения положительных и отрицательных (относительно избранной позиции) фактов, мнений, аргументов, точек зрения осуществляют расчет коэффициента Яниса (С). Если число положительных оценок превышает число отрицательных, коэффициент Яниса рассчитывается по формуле 2:



где f – число положительных оценок; n – число отрицательных оценок;

r – объем содержания, имеющего прямое отношение к изучаемой проблеме; t – общий объем анализируемого текста.

Если число положительных оценок меньше, чем отрицательных, коэффициент Яниса определяется по формуле 3:



При обработке полученных данных используются элементы ранжирования, теории вероятностей, матричного представления данных. Результаты контент-анализа чаще всего представляются графиками, рядами диаграмм (столбчатых или круговых). Для отображения отношений между единицами контент-анализа применяются также стандартные средства отображения структур, различные графы.

3. Интерпретация и синтезирование результатов осуществляются в соответствии с задачами конкретного исследования; выявляются наиболее очевидные и уникальные итоги работы, сравниваются ожидаемые и полученные данные [2].

Для web-разработок контент-анализ проводится для оценки: читабельности; выявление ошибок в тексте; оформления контента на странице сайта; адекватности иллюстраций на странице сайта; ссылки на странице по темам; содержание страницы.

1. Читабельность предполагает наличие статей на сайте с доступным для понимания текстом, а также оформление этих статей. Рекомендуется использовать простые предложения на сайте, они способствуют лучшему пониманию информации и делают текст понятным.

2. Проверка орфографии и пунктуации является залогом успеха сайта и репутации в целом.

3. Оформление контента является ключевым показателем успешности сайта. При формировании контента сайта необходимо помнить, что большой объем текстовой информации следует «разбавлять» иллюстрацией, а также текст следует выделять заголовками и подзаголовками. Только важно не перестараться, если текст сильно пестрит выделениями жирным, курсивом и прочими красотами – это отвлекает пользователя от чтения. Выделяйте только самые важные моменты.

4. Иллюстрация помогает лучше понять статью, но если иллюстрация не несет смысловую нагрузку для пользователя, то следует ее не использовать.

5. Оценка содержания проводится экспертом, который оценивает смысл текстовой информации, отслеживает отсутствие «воды».

С точки зрения поисковых машин оцениваются следующие параметры: релевантность контента поисковому запросу; «заспамленность» ключевыми словами; уникальность контента; объем контента; метатеги; видимость в поисковых системах.

Методы анализа документированной информации

Количественные методы анализа документированной информации. Библиометрия как комплекс количественных методов изучения документальных потоков. Сущность, назначение и область применения методов библиометрического анализа. Контент-анализ – количественный метод анализа качественных характеристик документов. Сущность и назначение контент-анализа. Область применения контент-анализа в библиотечно-информационной деятельности.

Вебометрия как средство информационной диагностики веб-пространства. Сущность и назначение методов вебометрического анализа. Использование вебометрического анализа при исследовании Интернет-ресурсов библиотечно-информационных учреждений.

Метод анализа ситуаций

Методы анализа ситуаций как инструмент оценки состояния предметной области. Состав и общая характеристика методов анализа ситуаций.

Сущность и назначение метода синтеза проблемных ситуаций. Структура ситуации проблемы. Использование метода синтеза проблемных ситуаций при проектировании автоматизированных библиотечно-информационных систем, электронных информационных ресурсов.

SWOT-анализ как специфический управленческий метод анализа информации, характеризующей положение организации на рынке. Сущность и назначение SWOT-анализа. Возможности применения SWOT-анализа при составлении стратегических планов развития АБИС, продвижения информационных продуктов и услуг на рынок.

Моделирование как инструмент анализа предметной области Классификация и общая характеристика методов моделирования, применяемых при анализе предметных областей.

Модели информационных объектов и процессов. Характеристика средств моделирования предметной области документальных и фактографических информационных систем.

Концептуальная модель как средство формализованного представления структуры предметной области. Модель «сущность-связь» (ER-модель) – важнейшее концептуальное средство описания предметной области. ER-диаграмма как способ представления структуры предметной области при

ER-моделировании.

Направления использования методов моделирования при проведении анализа предметных областей.

Форма представления результата:

Оформить отчет по практическому занятию.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход решения задачи должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при решении задачи и примеров допущено более 5 вычислительных ошибок.

Раздел 1. Основы проектирования информационных систем

Практическое занятие № 2

«Изучение устройств автоматизированного сбора информации» Цель: изучение устройств автоматизированного сбора информации

Выполнив работу, Вы будете:

уметь:

- У3. осуществлять выбор модели и средства построения информационной системы и программных средств;
- У4. использовать алгоритмы обработки информации для различных приложений;
- У9. работать с инструментальными средствами обработки информации.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

1. Изучить технологию штрихового кодирования; технологию радиочастотной идентификации (RFID); карточную технологию (CardTechnologies); технологию сбора данных (DataCommunicationsTechnologies).

Порядок выполнения работы:

1. Сделать конспект тем: технология штрихового кодирования (BarCodeTechnologies); технологии радиочастотной идентификации (RFID – RadioFrequencyIdentificationTechnologies); карточная технология (CardTechnologies); технология сбора данных (DataCommunicationsTechnologies).
2. Продумать для каких сфер деятельности человека следует использовать: технологию штрихового кодирования; технологию радиочастотной идентификации (RFID); карточную технологию (CardTechnologies); технологию сбора данных (DataCommunicationsTechnologies). Ограничением выступает наличие технических средств. Сферы деятельности: транспортная; информационная; экономическая.
3. Для заданной предметной области (см. практическая работа № 1) опишите устройства и методы автоматизированного сбора информации.

Ход работы:

Технология штрихового кодирования (BarCodeTechnologies)

Технология штрихового кодирования (Bar Code Technologies) - сегодня самая известная из

всех технологий бесконтактной идентификации. В соответствии с ней для эффективного учета движения материальных ценностей каждому товару присваивают уникальный код и обеспечивают его быстрое считывание при минимальных ошибках.

Штриховое кодирование было изобретено американским инженером Давидом Коллинзом, который после окончания в 1950-х гг. инженерного факультета Массачусетского технологического института, поступил работать на Пенсильванскую железную дорогу, где ему пришлось столкнуться с проблемой сортировки вагонов. Чтобы упростить распознавание вагонов он предложил записывать их номера не только обычными цифрами, но и специальным кодом, состоящим из красных и синих полос, расположенных на стенке вагона в прямоугольнике длиной до полуметра.

Испытания подтвердили, что сканирующее устройство способно правильно считывать коды даже при скорости движения вагона около 100 км/час. В 1968 г. для этой цели впервые использовали лазерный луч. Штриховой код - это символ, состоящий из рисунка полос (штрихов) и пространства между ними (пробелов), отображающий машинный код букв и чисел в двоичной системе. Штрих (полоса) - темная зона изображения на однотонном светлом фоне, ограниченная прямыми параллельными линиями или концентрическими окружностями. Элементы штрихового кода наносятся на поверхность носителя, имеющего определенные светотехнические характеристики. При этом штрихи, наносимые с помощью красителей или каких-то других средств, хорошо поглощают свет на определенных длинах волн, а фоновая поверхность хорошо его отражает, что и используется при оптическом считывании.

Пробел - пространство между штрихами. В большинстве кодов в ширине пробела заключена определенная информация, лишь в некоторых кодах пробел - вспомогательная часть изображения и выполняет функцию элемента-разделителя. Штриховое кодирование. У штриховых кодов существует множество различных кодировок. Каждая из них имеет свои собственные правила для изображения символа, т. е. написание, порядок слов, знаки препинания, требований для печати и декодирования, проверки ошибок и других характеристик. Различные кодировки отличаются как по представлению данных, так и по типам данных, которые они могут содержать: некоторые кодируют только цифры, другие - цифры, буквы и некоторые знаки препинания. Новейшие кодировки имеют возможность кодировать символы из различных языков одновременно, а некоторые даже позволяют с помощью преднамеренно заложенной избыточности восстанавливать закодированные данные, если вдруг код поврежден. Наиболее широко используются так называемые линейные штриховые коды. Они состоят из темных штрихов и светлых пробелов между ними, соотношение ширины которых и определяет закодированную информацию. Эти коды могут содержать обычно от 15 до 50 символов в зависимости от типа и формы.

Двумерные штриховые коды разработаны для повышения количества кодируемой информации. В зависимости от типа максимальное количество содержащихся символов может составлять до 2000, а у некоторых - почти 4000. Двумерные штриховые коды подразделяются на два основных вида: многорядные коды (англ. multi-row code) и матричные коды (matrix code). В многорядных кодах данные кодируются в виде нескольких строчек обычных одномерных штриховых кодов. Они находятся одна над другой и составляют форму прямоугольника, но при этом содержат единое информационное сообщение. Это как единый текст, написанный в несколько строк. В отличие от традиционных линейных символик штрихового кода, которые позволяют представлять в символе штрихового кода короткую последовательность данных, являющуюся, как правило, ключом к записи во внешней базе данных, многострочные символики позволяют кодировать информацию в полном объеме. Кроме того, многострочные символики включают в себя специальные механизмы по сжатию данных (защите их от повреждения, связыванию информации), представленных в нескольких символах, в один большой файл; представлению различных наборов знаков в одном сообщении. Примерами таких кодов являются PDF 417, MaxiCode. Матричный код основан на расположении черных элементов внутри матрицы. Каждый черный элемент имеет одинаковый размер, и данные кодируются позицией элемента.

Матричные коды обеспечивают максимально возможную плотность информации, которую можно считать оптическими методами. По форме они бывают квадратными, шестиугольными и круглыми. Эти коды считываются только с помощью специализированных сканеров изображений со встроенными декодерами. Их особенность состоит в том, что эти коды могут быть изготовлены

не только печатью на этикетках, но и, например, гравировкой и штамповкой на металле и других материалах. Примерами таких кодов являются Data Matrix, Aztec Code.

В настоящее время все более широко используется еще одно семейство кодов - так называемые композитные символика (Composite Symbolologies). Композитная символика состоит из двух частей: линейного символа и напечатанного над ним двумерного компонента. В этом семействе два кода располагаются на фиксированном расстоянии друг от друга и содержат взаимосвязанную информацию. Они предназначены для задач, в которых в различные моменты времени нужны разные виды информации о кодируемом объекте. Отличительной особенностью композитной символика является использование линейного символа в качестве ссылки (ключевой информации) для двумерного компонента. Это позволяет существенно сократить площадь двумерного компонента. Стандарты штриховых кодов распространяются на их печать, сканирование и верификацию. Эти стандарты обеспечивают взаимодействие производителей оборудования и этикеток и тех, кто использует коды в своей деятельности.

Изготовление штриховых кодов осуществляется двумя способами - печать на рабочем месте самим пользователем и печать производителем этикеток. В первом случае содержащиеся в коде данные обычно переменные, вводимые самим пользователем с клавиатуры или загружаемые с компьютера из базы данных. При этом наиболее часто используются следующие технологии печати: термоспособ, термоперенос (термотрансферная печать), матричная, струйная и лазерная. Принтеры штриховых кодов могут оснащаться рядом дополнительных устройств, таких как автоматические аппликаторы этикеток для наклеивания большого объема этикеток с изменяющейся информацией, и беспроводное сетевое оборудование для мобильных приложений и работы в полевых условиях.

Технологии радиочастотной идентификации (RFID RadioFrequencyIdentificationTechnologies)

Что такое RFID?

RFID представляет собой метод автоматической идентификации через радиосигнал. Система состоит из считывателей, меток и программного обеспечения. Метка – это микросхема, в которой хранятся данные, а также антенна для беспроводной передачи информации. Внешний считыватель сканирует память метки радиочастотной идентификации и обрабатывает полученные данные. Программное обеспечение отвечает за целостную работу системы.

Способ автоматической идентификации объектов с каждым годом используется все чаще. С помощью Radio Frequency IDentification проводят сверку наличия имущества организации и состояния финансовых обязательств, автоматизируют производственные процессы, устанавливают подлинность объектов, отслеживают логистику поставок. Постараемся разобраться детальнее, как работает система и в чем ее преимущество.

Принципы работы RFID

Radio Frequency IDentification по сути является усовершенствованным алгоритмом радиолокационного опознавания «Свой-чужой». Аппаратно-программный технический комплекс для автоматического поиска отличий использует радиоволны определенной частоты. Данные хранятся в специальной метке, которую можно принимать, а затем дешифровать удаленно с помощью специальных считывателей.

Технология имеет достаточно простой механизм работы:

- информация записывается с помощью радиоволн на микрочип;
- данные поступают на считыватель с помощью радиосигнала встроенной антенны;
- определение излучаемой частоты, настройка и считывание сведений осуществляется автоматически за счет сканера.

Спрос на РЧИ появился еще в начале двадцать первого века. Система постоянно усовершенствуется, используется для маркировки и автоматизированной идентификации продукции. Но перечень функций RFID гораздо шире. С помощью современной технологии

можно выполнить идентификацию и контролировать движение маркированных объектов – как живых, так и неживых.

Основные преимущества RFID

По сравнению с QR-кодом, радиочастотная идентификация обладает следующими преимуществами:

- метки практически невозможно подделать, предусмотрена функция перезаписи неограниченное количество раз;
- допускается интеграция микрочипов в любую часть объекта, стандарты European Article Number не используются;
- микросхема вмещает до десяти гигабайт данных о предмете;
- непосредственный контакт или прямая видимость не обязательны;
- идентификация проводится на расстоянии около трех сотен метров;
- комплекс работает бесперебойно от 10 лет;
- интеллектуальные решения принимаются в режиме онлайн;
- технология может использоваться в самых агрессивных условиях, а метки читаются через краску, грязь, воду, пар, древесину;
- групповой учет нескольких движущихся объектов – до двухсот меток.

Данные на метке можно засекретить. Закрывается доступ полностью или только часть информации.

Определение RFID метки

Метка RFID также известна как транспондер. В микроустройство встроен чип, на котором хранится персональный номер и другие данные пользователя. Антенна принимает сигналы считывателя и передает информацию ему.

Классификация RFID-меток

По типу конструкции и материалу маркировки метки делятся на следующие категории:

- интегрированные внутрь;
- клеящиеся;
- встроенные вовнутрь корпуса.

По источнику питания транспондеры бывают:

1. Пассивными. Самый простой вариант с микрочипом и антенной. Активация, получение питания и передача импульса возможна лишь при попадании считывателя в зону электромагнитного излучения. Радиус действия электромагнитного поля считывателя, для активации и работы, составляет 0,5-8,0 метров.

2. Полупассивные транспондеры включают антенну, микрочип и батарею питания. Активация происходит за счет действия импульса считывателя. Данные передаются посредством собственного питания. Дальность работы таких устройств больше, но во многом зависит от мощности считывателя и чувствительности антенны.

3. Самостоятельную и стабильную трансляцию сигнала обеспечивают активные метки. Они включают антенну, микрочип и источник питания. Этот вариант дороже остальных, имеет ограниченный срок годности емкости батареи, но и в то же время отличается явным преимуществом – гораздо большим объемом памяти. В комплект могут входить дополнительные датчики влажности, температуры, загазованности, радиации. Передача сигнала осуществляется на расстояние до 300 метров.

По типу памяти различают следующие метки:

- RO (одноразовые) — данные записываются однократно, используются, как правило, для идентификации;
- WORM (для однократного прочтения) — информацию записывают один раз, но считывать можно без ограничений;
- RW (для записи и перезаписи) – сведения можно несколько раз считывать и перезаписывать.

Делятся транспондеры и по частоте, на которой происходит передача закодированных данных:

1. Метки с наибольшей дальностью действий называют сверхчастотными. Диапазон значений: от 860 до 960 мегагерц. Изначально были разработаны для упрощения организации складского хозяйства.

2. Сравнительно недорогие и экологически безопасные теги 13,56 мегагерц нашли свое применение в ПС и логистике. Их устанавливают в карты для проезда в метро и другом общественном транспорте.

3. Низкочастотные теги используют для чипирования людей и животных. Зона действия от 125 до 164 кГц не позволяет считать данные на большом расстоянии.

Карточная технология (CardTechnologies)

Карточные технологии (Card Technologies) делятся на три класса: технологии на основе магнитной полосы, смарт-карты, оптической карты. Первая карточка с магнитной полосой появилась в 1960-х гг. на проездных билетах, а в 1970-х гг. - на банковских карточках. С того времени области применения карточек с магнитной полосой продолжает расти. Однако магнитная полоса ограничена по объему информации, которая может быть записана на нее, также остро стоит вопрос надежности считывания и безопасности данных. С появлением новых технологий обсуждается вопрос о целесообразности развития карт с магнитной полосой. В ближайшее время эта технология будет существовать, так как она глубоко внедрилась в жизнь общества, и обеспечивает недорогие массовые технические решения. Смарт-карта (другие распространенные названия - чип-карта, интегрированная карта) представляет собой предмет размером с пластиковую кредитную карту, в котором размещена интегральная микросхема для хранения информации.

Принято различать пассивные смарт-карты, другое название - «молчаливые» и активные смарт-карты, другое название - «умные», интеллектуальные. Смарт-карты первого типа содержат только микросхему памяти и используются только для хранения информации. Второй тип смарт-карт содержит наряду с микросхемой памяти - микропроцессор. В этом случае карта имеет возможность принимать решения о хранящейся информации и обеспечивать различные методы для защиты доступа к информации. Именно безопасность в свое время рассматривалась как основная причина замены других технологий смарт-картой. Смарт-карта, содержащая микропроцессор, также делится на два вида: контактная и бесконтактная. Оба вида имеют встроенный микропроцессор, однако последняя не имеет контактов, покрытых золотом. Она использует технологии обмена информацией между картой и считывающим устройством без какого-либо физического контакта, ее преимуществом является больший срок службы, для нее исключена возможность уничтожения информации в процессе считывания. Хотя в последнее время цены на смарт-карты значительно упали, но все равно они достаточно высоки по сравнению с картами с магнитной полосой. Самым большим преимуществом смарт-карт является большой объем информации, который может быть записан на ней и безопасность информации, которую также обеспечивает карта.

Первое упоминание о смарт-картах появилось во Франции в 1974 г., в практику они были также внедрены во Франции в 1982 г. Эта технология очень быстро распространялась и принималась в Европе. Карты с оптической памятью основаны на том же принципе, что и музыкальные диски и СЭ-ЯОМ. На карту прикрепляется лазерная панель, покрытая золотом, и она используется для хранения информации. Материал, используемый для этой панели, состоит из нескольких слоев и активизируется, когда на них попадает лазерный луч. Лазер выжигает крошечное отверстие в этом материале, которое потом будет различаться в процессе считывания. Наличие или отсутствие таких выжженных точек означает «единица» или «ноль». Оптическая карта может хранить информацию объемом от 4 до 6,6 Мб.

Технология сбора данных (DataCommunicationsTechnologies)

Сбор предполагает получение максимально выверенной исходной информации и является одним из самых ответственных этапов в работе с информацией, поскольку от цели сбора и методов последующей обработки полностью зависит конечный результат работы всей информационной системы.

Технология сбора подразумевает использование определенных методов сбора информации и технических средств, выбираемых в зависимости от вида информации и применяемых методов ее

сбора. На заключительном этапе сбора, когда информация преобразуется в данные, т. е. в информацию, представленную в формализованном виде, пригодном для компьютерной обработки, осуществляется ее ввод в систему.

Хранимые данные должны быть в достаточном объеме доступны для извлечения из места хранения, отображения, передачи или обработки по запросу пользователя. Сбор данных должен обеспечивать необходимую полноту и минимальную избыточность хранимой информации, что может быть достигнуто за счет выбора данных, оценки их необходимости, а также анализа существующих данных и разделения их на входные, промежуточные и выходные.

Входные данные — это данные, получаемые из первичной информации, создающие исходное описание предметной области и подлежащие хранению.

Промежуточные данные формируются из других данных в процессе преобразований и обработки, и, как правило, не подлежат длительному хранению.

Выходные данные есть результат обработки входных данных по соответствующему алгоритму; они служат основанием для принятия управленческих решений и подлежат хранению в течение определенного срока.

Для сбора данных необходимо сначала определить технические средства, позволяющие осуществлять сбор быстро и высококачественно и поддерживающие операции ввода информации и представления данных в электронной форме, которые выбираются в зависимости от типа собираемой информации и ее назначения (сканер, сенсорный экран, диктофон, видеокамеры и фотоаппараты и др.).

В промышленных системах в зависимости от сферы применения используются технические средства для сканирования штрих-кода, захвата изображений, автоматические датчики объема, давления, температуры, влажности, системы распознавания сигналов и кодов и т.д. В целом применение подобных промышленных средств сбора информации называют технологией автоматической идентификации, т. е. прямым сбором данных в микропроцессорное устройство (компьютер или программируемый контроллер) без использования клавиатуры. Такая технология применяется для исключения ошибок, связанных со сбором данных, и ускорения процесса сбора; она позволяет не только идентифицировать объекты, но и следить за ними, кодировать большое количество информации

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 1. Основы проектирования информационных систем

Практическое занятие № 3

«Оценка экономической эффективности информационной системы»

Цель: изучение методов оценки экономической эффективности информационных систем.

Выполнив работу, Вы будете:

уметь:

- У3. осуществлять выбор модели и средства построения информационной системы и программных средств;
- У4. использовать алгоритмы обработки информации для различных приложений.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

1. По выбранным предметным областям (Практическое занятие 1) спроектировать информационную систему. Рассчитать срок окупаемости информационной системы.
2. Составить сравнительный анализ методов оценки экономической эффективности информационных систем

Порядок выполнения работы:

Метод расчета срока окупаемости

Метод расчета срока окупаемости зависит от равномерности распределения прогнозируемых доходов от инвестиций: если доход распределен по годам равномерно, то срок окупаемости рассчитывается делением единовременных затрат на величину годового дохода, обусловленного ими; если прибыль распределена неравномерно, то срок окупаемости рассчитывается прямым подсчетом числа лет, в течение которых инвестиция будет погашена кумулятивным доходом. Расчет окупаемости представляет собой анализ возврата средств исходя из принятых в компании максимальных сроков окупаемости вложений.

Формула срока окупаемости:

$$PP = \frac{I_0}{CF_{CF}}, \quad (7)$$

где PP – срок окупаемости инвестиций (лет);

I_0 – первоначальные инвестиции;

CF_{CF} – среднегодовая стоимость поступлений от реализации проекта.

Такое определение срока окупаемости возможно при равномерном поступлении денежных средств по годам.

В противном случае:

$$T_{OK} = n, \quad \text{при котором} \quad \sum_{t=1}^n CF_t > I_0, \quad (8)$$

где T_{OK} – срок окупаемости инвестиций;

n – число периодов;

CF_t – приток денежных средств в период t ;

I_0 – величина исходных инвестиций в нулевой период.

$$T_{OK} = n, \quad \text{при котором} \quad \sum_{t=1}^n CF_t > I_0,$$

где T_{OK} – срок окупаемости инвестиций;

n – число периодов;

CF_t – приток денежных средств в период t ;

I_0 – величина исходных инвестиций в нулевой период.

Метод расчета срока окупаемости не учитывает временной стоимости денег. Метод позволяет определить, сколько необходимо времени, чтобы инвестиции принесли столько денежных средств, сколько было потрачено. Помимо этого, данный метод не учитывает поступление денежных средств после окупаемости проекта, то есть прибыль проекта не считается.

Пример расчета срока окупаемости информационной системы для инвентаризации. Нормативный срок окупаемости не должен превышать 3 лет [3].

Проектные затраты на разработку системы определяются по формуле:

$$Z_{\text{проект}} = Z_{\text{об}} + Z_{\text{з/н}} + Z_{\text{ЕСН}} + Z_{\text{эн}} + Z_{\text{н.р}} + Z_{\text{ам}} + Z_{\text{пр}}$$

где $Z_{\text{об}}$ – затраты на приобретение оборудования, тг.;

$Z_{\text{з/н}}$ – заработная плата исполнителям и разработчикам проекта, тг.; $Z_{\text{ЕСН}}$ – отчисления на единый социальный налог, тг.;

$Z_{\text{эн}}$ – затраты на электроэнергию, тг.; $Z_{\text{н.р}}$ – накладные расходы, тг.;

$Z_{\text{ам}}$ – затраты на амортизацию, тг.;

$Z_{\text{пр}}$ – прочие затраты, тг.

В таблице 1 представлены затраты на оборудование (по списку оборудования). Таблица 1.1 - Список необходимого оборудования и его стоимость.

Наименование	Кол-во	Затраты (тг)
Системный блок (Core2Duo-E7400 2.6GHz, 2048Mb, HDD 500Gb)	1 шт	108250
Монитор 19 «Samsung Sync Master 943BM)	1 шт	39600
Клавиатура GeniusSlimStar 100	1 шт	1250
Мышь Genius NetScroll 110 white optical	1 шт	800
МФУ Epson Stylus Photo R390	1 шт	34550
DVD-RW NEC (Optiarc) AD5200S-01 beige	1 шт	3600
Итого:		188050

Транспортно-заготовительные расходы:

расход проектирование информационный система $188050 \cdot 0.1 = 18805$ тг.

Итого: $Z_{\text{об}} = 188050 + 18805 = 206855$ тг.

Затраты на заработную плату исполнителям и разработчикам проекта

Для этого необходимо рассчитать ориентировочные трудозатраты и стоимость человеко-часов.

$$Z_{\text{з/н}} = \sum_i^n T_{\text{opi}} C_{\text{ч/чj}}$$

– заработная плата j-го исполнителя;

– ориентировочные трудозатраты i-го этапа, j-го исполнителя

– стоимость человека-часа j-го исполнителя.

$$Z_{\text{з/н}} = \sum_i^n T_{\text{opi}} C_{\text{ч/чj}}$$

$$= 45000/22*8 = 255,68 \text{ тг.}$$

Средняя заработная плата инженера-проектировщика составляет 30000 тг. в месяц, тогда:

$$= 30000/22*8 = 170,45 \text{ тг.}$$

Таблица 1.2 – Программа проведения проекта.

Этапы работы	Исполнитель	Ориентировочные трудозатраты чел.-час (Тор)	Стоимость, чел.-час	Затраченные дни
Постановка задачи	Руководитель	16	255,68	2
Определение хода работы	Инженер-проектировщик	8	170,45	1
	Руководитель	8	255,68	1
Анализ поставленной задачи	Инженер-проектировщик	8	170,45	1
	Руководитель	8	255,68	1
Изучение объекта автоматизации	Инженер-проектировщик	56	170,45	7
	Руководитель	16	255,68	2
Составление плана работы	Инженер-проектировщик	8	170,45	1
	Руководитель	8	255,68	1
Согласование и корректировка плана работы	Инженер-проектировщик	8	170,45	1
	Руководитель	8	255,68	1
Описание и характеристика исследуемого объекта автоматизации	Инженер-проектировщик	120	170,45	15
Оформление проекта	Инженер-проектировщик	40	170,45	5
Проверка проекта	Руководитель	16	255,68	2
Итого:		328		41

Таблица 1.3 – Расчёт заработной платы.

Должность исполнителя	Месячный оклад, тг.	Стоимость человеко-часатг.	Трудо- затраты, чел. час.	Зарплата по окладу, тг.
Инженер- проектировщик	30000	170,45	248	42271,6
Руководитель проекта	45000	255,68	80	20454,4
Консультант проекта по экономической части	35000	198,86	16	3181,76
Консультант по охране труда	35000	198,86	16	3181,76

Итого				69089,52
-------	--	--	--	----------

Итого: $Z_{з/п} = 69089,52$ тг.

Отчисления на ЕСН

Единый социальный налог принять в размере 20%. $Z_{ЕСН} = Z_{з/п} * 0,20$

$Z_{ЕСН} = 69089,52 * 0,20 = 13817,904$ тг.

Затраты на электроэнергию

Затраты на электроэнергию по каждому токоприемнику рассчитываются по формуле: $Z_{эн} = P_i * T_{фi} * Ц_{ол} * K_i$

где P_i - потребляемая мощность i-ого токоприемника, кВт/ч; $T_{фi}$ - фактическое время работы i-ого токоприемника, тг; $Ц_{ол}$ - цена на электроэнергию за 1 кВт/ч;

K_i - коэффициент использования по времени i-ого токоприемника, %. Таблица 1.4 - Расчета затрат на электроэнергию.

Наименование токоприёмника	Мощность, Вт	Время работы, ч	Стоимость, тг/кВт-ч	Кэф-т использования	Затраты
Компьютер	250	100	15	0,9	337,5
Принтер	100	100	15	0,1	15
Освещение	300	100	15	0,6	270
Итого					622,5

Накладные расходы

Накладные расходы в размере 10% от заработной платы исполнителей: $Z_{н.р} = 69089,52 * 0,1 = 6908,95$ тг.

Расходы на амортизацию

Расходы на амортизацию составляют 15% от стоимости оборудования: $Z_{ам} = 206855 * 0,15 = 31028,25$ тг.

Затраты на этапе проектирования.

Таблица 1.5 – Смета затрат.

п/п	Наименование затрат	Сумма, тг.
1	Расходы на оборудование (Зоб)	206855
2	Расходы на заработанную плату (Зз/п)	69089,52
3	Отчисление на ЕСН (ЗЕСН)	13817,904
4	Расход на электроэнергию (Зэн)	622,5
5	Накладные расходы (Знр)	6908,95
6	Расходы на амортизацию (Зам)	31028,25
7	Прочие расходы (Зпр)	9849,66
	Итого	337586,604

Итого: $Z_{проект} = 338171,78$ тг.

Затраты на приобретение оборудования

Тоб – транспортные расходы (принимаем 10% от цены приобретения) тг;

Моб – расходы, связанные с монтажом оборудования (принимаем 5% от цены приобретения)тг.

Таблица 1.6 – Список необходимого оборудования и его стоимость.

Наименование	Кол-во	Цена за шт. (тг)	Затраты (тг)
Системный блок (Core2Duo-E7400, 2048Mb, 320Gb, 512Mb)	1 шт	95000	95000
Монитор 19 « ViewSonic VA903B-3 black «	1 шт	41500	41500
Клавиатура Genius KB-120 bl	1 шт	1250	1250
Мышь Genius NetScroll 310 silv Optical	1 шт	800	800
Кабель UTP 5e category Taiwan	1 шт	1700	1700
			140250

Тоб = $140250 * 0,1 = 14025$ тг.

Моб = $140250 * 0,05 = 7012$ тг.

Итого:

Зоб = $140250 + 14025 + 7012 = 161287$ тг.

Затраты на ремонт и содержание оборудования

Затраты на ремонт и содержание оборудования берем в размере 2,5% от стоимости основных средств:

Зрем = $161287 * 0,025 = 4032,17$ тг.

Таблица 1.8 – Смета затрат.

п/п	Наименование затрат	Сумма, тг.
	Расходы на оборудование (Зоб)	161287
	Расходы на ремонт (Зрем)	4032,17
	Расходы на заработную плату (Зз/п)	9545,2
	Отчисление на ЕСН (ЗЕСН)	1909,04
	Расходы на электроэнергию (Зэл)	3384,37
	Расходы на амортизацию (Зам)	24193,05
	Накладные расходы (Зн.р)	954,52
	Прочие расходы (Зпр)	6159,16
	Итого	211464,51

Итого: Зэкс = 211464,51 тг.

Экономическая эффективность

Для расчета экономического эффекта используем формулу:

$$\mathcal{E} = \text{Робщ} - \text{Ен} * \text{Кп}$$

где Робщ - годовая экономия;

Ен - нормативный коэффициент ($\text{Ен} = 0,15$);

Кп - капитальные затраты на проектирование и внедрение

В качестве годовой экономии от внедрения будет время сэкономленное сотрудниками переведенной в денежный показатель.

Расходы на персонал определим по формуле:

$$Z = n_s * z_s * (1 + A_c / 100)$$

где n_s – средняя численность персонала работа, которого будет автоматизирована; z_s – средняя заработная плата в месяц;

A_c - процент отчислений на социальное страхование; Данные, принимаемые для расчета:

Средняя заработная плата: 55 000 тг. Количество пользователей: 2 человек. Расходы на персонал:

1. В месяц:

$$Z_{\text{мес.}} = 2 * 55000 * (1 + 20 / 100) = 121\,000 \text{ тг.};$$

2. В год:

$$Z_{\text{год}} = Z_{\text{мес.}} * 12 = 1\,452\,000 \text{ тг.};$$

Расчет экономии за счет увеличения производительности труда пользователя. Расчет показателя повышения производительности труда произведен по формуле:

$$P = \left(\frac{\Delta T}{F - \Delta T} \right) * 100$$

где F - время, которое планировалось пользователем для выполнения работы до внедрения программ.

ΔT - экономия времени после внедрения программ. Оценка времени работы пользователей

При расчете сделаны следующие допущения:

На 6 типов операций из таблицы на следующем слайде каждый пользователь тратит 80% рабочего времени.

Фонд рабочего времени в месяц составляет 9 600 минут. Доля каждой операции в месяц одинакова

Все отделы проводят одинаковое время при работе с операциями.

Срок полезного использования программы – 7 лет, норма амортизации – 15% в год. Таблица 1.9

№ п/п	Вид работ	Среднее время на операцию в месяц на одного сотрудника, минут		Экономия времени в
				месяц, минут
		До автоматизации	После автоматизации	ΔT
1	Ввод информации	240	120	120
2	Проведение расчетов	150	60	90
3	Подготовка и печать отчетов	400	240	120
4	Анализ и выборка данных	1 500	750	750
5	Заполнение сложных документов по основной деятельности	1 500	750	750

6	Работа с электронными документами по основной деятельности	4 000	2 400	1 600
	Итого	7 790	4 320	3 470

Рассчитаем экономию:

$$P = 3\,470 / 7\,790 = 44,5\%$$

$$Z_{\text{год}} = Z_{\text{мес.}} * 12 = 1\,452\,000 \text{ тг.};$$

$$\mathcal{E} = Z_{\text{год.}} * P - E_{\text{п}} * K_{\text{п}} = 1\,452\,000 * 44,5\% - 0,15 * (337586,604 + 211464,51) = 563\,783 \text{ тг}$$

Экономический эффект от внедрения составляет 563 783 тг в год.

$$\mathcal{E}_{\text{год}} = \mathcal{E}_{\text{экс.анал.сист.}} - \mathcal{E}_{\text{экс.проект.сист.}}$$

Экономическая эффективность внедрения данной системы достигается за счёт увеличения количества производимых инвентаризаций и технического обслуживания.

$$\mathcal{E}_{\text{год}} = 53,25 * 8700 = 463275 \text{ тг.}$$

$$\text{Срок окупаемости } Z_{\text{общ}} = Z_{\text{проект}} + Z_{\text{экс}} = 85630,43 + 64575,56 = 150205,99 \text{ тг.}; T = Z_{\text{общ}} / \mathcal{E}_{\text{год}} = 150205,99 / 463275 = 0,32 \text{ г.}$$

$$549051,114 / 646140 = 0,85 \text{ г.}$$

где Т – срок окупаемости, лет;

Z_{общ} - суммарные затраты на разработку и внедрение системы, тг. Э_{год} – годовая экономия, тг.

Таблица 1.10. Техничко-экономические показатели проекта

Наименование показателя	Единица измерения	Значения показателя
Проектные затраты		
Затраты на оборудование (Зоб)	тг.	41371
Затраты на заработанную плату (Зз/п)	тг.	26867,36
Отчисление на ЕСН (ЗЕСН)	тг.	6985.51
Затраты на электроэнергию (Зэн)	тг.	54,37
Затраты на амортизацию (Зам)	тг.	2686,73
Накладные расходы (Знр)	тг.	5171,37
Прочие расходы (Зпр)	тг.	2494,09
Итого		85630,43
Эксплуатационные затраты		
Расходы на оборудование (Зоб)	тг.	33099
Расходы на ремонт (Зрем)	тг.	827,47
Расходы на заработную плату (Зз/п)	тг.	17417,3
Отчисление на ЕСН (ЗЕСН)	тг.	4528,5

Расходы на электроэнергию (Зэл)	тг.	943,35
Расходы на амортизацию (Зам)	тг.	4137,37
Накладные расходы (Зн.р)	тг.	1741,73
Итого	тг.	64575,56
Всего затрат	тг.	150205,99
Экономическая эффективность	тг.	401625
Срок окупаемости	лет	0,44

Ход работы:

Структурный анализ

В основе структурного анализа лежит выявление структуры как относительно устойчивой совокупности отношений, признание методологического примата отношений над элементами в системе, частичное отвлечение от развития объектов.

Структурный элемент является основным элементом структурного анализа. Структурный элемент по-другому называют структурным объектом. Структурный элемент предназначен для выполнения одного из элементарных функций, связанных с моделируемым предметом, процессом или явлением.

Структурный анализ предполагает исследование предметной области с помощью графического представления модели системы. Данная система рассматривается как один большой объект, который в дальнейшем детализируется, приобретая иерархическую структуру.

Для такого подхода характерны:

- разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 9);
- указание сущности каждого уровня системы;
- использование строгих формальных правил записи;
- последовательное приближение к конечному результату.

Цель структурного анализа – это переход от абстрактных, общих знаний об исследуемой предметной области в точную модель.

Процесс декомпозиции является условным приемом, позволяющим представить систему в виде, удобном для восприятия, и оценить ее сложность. В результате декомпозиции подсистемы по определенным признакам выделяются отдельные структурные элементы и связи между ними. Процесс декомпозиции представляет собой разбиение одной большой системы на более мелкие подсистемы. Например, существует система «Колледж». В процессе декомпозиции можно выделить ряд подсистем, такие как отделения, рисунок 1.



Рисунок 1 – декомпозиция системы «Колледж»

Контент-анализ

Контент-анализ — метод количественного анализа содержания исследуемых текстов, применяемый для измерения в них каких-либо количественно определяемых характеристик.

В информационно-аналитической деятельности измеряются: частота ключевых слов и словосочетаний, служащих для описания объекта; повторяемость избранных объектов (имен, наименований фирм, марок продукции и др.) в некотором наборе документов/сайтов, объемы текстов; сила связи между публикациями по признакам социотирования; оценивается вклад научных коллективов в разработку объекта.

База исследования может включать вторичные или первичные документы разного целевого назначения: материалы конференций, обзорные публикации об объекте через какие-либо промежутки времени, новостные сообщения СМИ, поток правовых актов и комментариев к ним, рекламных материалов, авторефератов диссертаций.

Отбор единиц анализа основывается на семантических критериях. **Категории анализа** — это обобщающие ключевые слова и словосочетания, имеющиеся в тексте и соответствующие теме исследования.

В качестве единиц анализа принято считать: термины, понятия, ключевые слова; выраженные словами или словосочетаниями; темы, проблемы, тенденции развития; имена людей, названий организаций, географические названия, имена собственные.

Единица счета представляет собой количественную меру единицы анализа, которая регистрирует частоту появления признака категории анализа в тексте. Например, частота встречаемости единиц анализа в текстах, на сайтах, объем текстов, объем статьи, длительность трансляции по радио или ТВ; размер аудио- и видеозаписей и др.

Этапы контент-анализа:

4. Подготовительный этап. Включает определение цели, задач, гипотезы исследования; отбор источников информации и массива текстов для анализа, выбор категорий и единиц анализа, единиц счета, технологии обработки данных, разработка инструментов — форматов таблиц, систем кодирования, условных сокращений, инструкций аналитику и др. В числе обязательных инструментов рекомендуются:

- классификатор контент-анализа — общая таблица, в которую сведены все категории (подкатегории) анализа и единицы анализа; ее основное назначение — предельно четко зафиксировать те единицы, в которых выражается каждая категория, используемая в исследовании;

- протокол (бланк) контент-анализа — содержит сведения о документе (его авторе, времени издания, объеме и т. п.); итоги его анализа (количество случаев употребления в нем определенных единиц анализа и следующие отсюда выводы относительно категорий анализа); протоколы заполняются, как правило, в закодированном виде;

- регистрационная карточка представляет собой кодировальную матрицу, в которой отмечается количество единиц счета, характеризующее единицы анализа; протокол контент-анализа каждого конкретного документа заполняется на основе подсчета данных всех регистрационных карт, относящихся к этому документу;

- инструкция аналитику (кодировщику) — система правил и пояснений для исполнителей по сбору эмпирической информации, регистрации заданных единиц анализа.

5. Сбор и первичная обработка данных контент-анализа. Производится процедура подсчета семантических единиц разного уровня обобщения. Фиксация данных при контент-анализе обычно осуществляется с помощью достаточно простых анкет, в которых фиксируется каждое появление в анализируемом тексте искомой единицы. Составляются счетные таблицы по отдельным текстам, а также сводные таблицы по отдельным категориям (темам).

Самый простой вариант контент-анализа — количественный подсчет частоты встречаемости ключевых слов в тексте (по массиву в целом, по отдельным группам публикаций, по отдельным единицам наблюдения и др.). Также может быть определен удельный вес той или иной категории (К), который можно вычислить с помощью формулы 1:



Для установления соотношения положительных и отрицательных (относительно избранной позиции) фактов, мнений, аргументов, точек зрения осуществляют расчет коэффициента Яниса (С). Если число положительных оценок превышает число отрицательных, коэффициент Яниса рассчитывается по формуле 2:



где f – число положительных оценок; p – число отрицательных оценок;
 g – объем содержания, имеющего прямое отношение к изучаемой проблеме; t – общий объем анализируемого текста.

Если число положительных оценок меньше, чем отрицательных, коэффициент Яниса определяется по формуле 3:



При обработке полученных данных используются элементы ранжирования, теории вероятностей, матричного представления данных. Результаты контент-анализа чаще всего представляются графиками, рядами диаграмм (столбчатых или круговых). Для отображения отношений между единицами контент-анализа применяются также стандартные средства отображения структур, различные графы.

6. Интерпретация и синтезирование результатов осуществляются в соответствии с задачами конкретного исследования; выявляются наиболее очевидные и уникальные итоги работы, сравниваются ожидаемые и полученные данные [2].

Для web-разработок контент-анализ проводится для оценки: читабельности; выявление ошибок в тексте; оформления контента на странице сайта; адекватности иллюстраций на странице сайта; ссылки на странице по темам; содержание страницы.

6. Читабельность предполагает наличие статей на сайте с доступным для понимаем текстом, а также оформление этих статей. Рекомендуются использовать простые предложения на сайте, они способствуют лучшему пониманию информации и делают текст понятным.

7. Проверка орфографии и пунктуации является залогом успеха сайта и репутации в целом.

8. Оформление контента является ключевым показателем успешности сайта. При формировании контента сайта необходимо помнить, что большой объем текстовой информации следует «разбавлять» иллюстрацией, а также текст следует выделять заголовками и подзаголовками. Только важно не перестараться, если текст сильно пестрит выделениями жирным, курсивом и прочими красотами – это отвлекает пользователя от чтения. Выделяйте только самые важные моменты.

9. Иллюстрация помогает лучше понять статью, но если иллюстрация не несет смысловую нагрузку для пользователя, то следует ее не использовать.

10. Оценка содержания проводится экспертом, который оценивает смысл текстовой информации, отслеживает отсутствие «воды».

С точки зрения поисковых машин оцениваются следующие параметры: релевантность контента

поисковому запросу; «заспамленность» ключевыми словами; уникальность контента; объем контента; метатеги; видимость в поисковых системах.

Методы анализа документированной информации

Количественные методы анализа документированной информации. Библиометрия как комплекс количественных методов изучения документальных потоков. Сущность, назначение и область применения методов библиометрического анализа. Контент-анализ – количественный метод анализа качественных характеристик документов. Сущность и назначение контент-анализа. Область применения контент-анализа в библиотечно-информационной деятельности.

Вебометрия как средство информационной диагностики веб-пространства. Сущность и назначение методов вебометрического анализа. Использование вебометрического анализа при исследовании Интернет-ресурсов библиотечно-информационных учреждений.

Метод анализа ситуаций

Методы анализа ситуаций как инструмент оценки состояния предметной области. Состав и общая характеристика методов анализа ситуаций.

Сущность и назначение метода синтеза проблемных ситуаций. Структура ситуации проблемы. Использование метода синтеза проблемных ситуаций при проектировании автоматизированных библиотечно-информационных систем, электронных информационных ресурсов.

SWOT-анализ как специфический управленческий метод анализа информации, характеризующей положение организации на рынке. Сущность и назначение SWOT-анализа. Возможности применения SWOT-анализа при составлении стратегических планов развития АБИС, продвижения информационных продуктов и услуг на рынок.

Моделирование как инструмент анализа предметной области Классификация и общая характеристика методов моделирования, применяемых при анализе предметных областей.

Модели информационных объектов и процессов. Характеристика средств моделирования предметной области документальных и фактографических информационных систем.

Концептуальная модель как средство формализованного представления структуры предметной области. Модель «сущность-связь» (ER-модель) – важнейшее концептуальное средство описания предметной области. ER-диаграмма как способ представления структуры предметной области при ER-моделировании.

Направления использования методов моделирования при проведении анализа предметных областей.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 1. Основы проектирования информационных систем

Практическое занятие № 4

Разработка модели архитектуры информационной системы

Цель: получение навыков разработки модели архитектуры информационной системы.

Выполнив работу, Вы будете:

уметь:

- У3. осуществлять выбор модели и средства построения информационной системы и программных средств;
- У4. использовать алгоритмы обработки информации для различных приложений.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

1. Согласно заданному варианту (практическая работа 1) Продумать, какая архитектура информационной системы может быть применима к индивидуальному заданию (практическая работа 1). Архитектура может быть: централизованной; «файл-сервер»; «файл-клиент». Объяснить почему вы выбрали данную архитектуру.

Порядок выполнения работы:

1. Проанализировать методические рекомендации и выписать стандарты качества автоматизированных информационных систем.
2. Проанализировать методические рекомендации и выписать стандарты качества автоматизированных информационных систем.
3. Самостоятельно описать стандарты группы ISO, используя интернет источники.

Ход работы:

Определение архитектуры информационной системы [4]:

- Архитектура – это организационная структура системы.

- Архитектура информационной системы – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

- Архитектура – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

- Архитектура – это набор значимых решений по поводу организации системы программного обеспечения, набор структурных элементов и их интерфейсов, при помощи которых конструируется система, вместе с их поведением, определяемым во взаимодействии между этими элементами, компоновка элементов в постепенно укрупняющиеся подсистемы, а также стиль архитектуры, который направляет эту организацию – элементы и их интерфейсы, взаимодействия и компоновку.

- Архитектура программы или компьютерной системы – это структура или структуры системы, которые включают элементы программы, видимые извне свойства этих элементов и связи между ними.

- Архитектура – это структура организации и связанное с ней поведение системы. Архитектуру можно рекурсивно разобрать на части, взаимодействующие посредством интерфейсов, связи, которые соединяют части, и условия сборки частей. Части, которые взаимодействуют через интерфейсы, включают классы, компоненты и подсистемы.

- Архитектура программного обеспечения системы или набора систем состоит из всех важных проектных решений по поводу структур программы и взаимодействий между этими структурами, которые составляют системы. Проектные решения обеспечивают желаемый набор свойств, которые должна поддерживать система, чтобы быть успешной. Проектные решения предоставляют концептуальную основу для разработки системы, ее поддержки и обслуживания.

Классификация архитектур информационной системы

1. Централизованная архитектура

Централизованная архитектура вычислительных систем была распространена в 70-х и 80-х годах и реализовывалась на базе мейнфреймов (например, IBM-360/370 или их отечественных аналогов серии ЕС ЭВМ), либо на базе мини-ЭВМ (например, PDP-11 или их отечественного аналога СМ-4). Характерная особенность такой архитектуры – полная "неинтеллектуальность" терминалов. Их работой управляет хост-ЭВМ.

Достоинства такой архитектуры:

- пользователи совместно используют дорогие ресурсы ЭВМ и дорогие периферийные устройства;
- централизация ресурсов и оборудования облегчает обслуживание и эксплуатацию вычислительной системы;
- отсутствует необходимость администрирования рабочих мест пользователей;

Главным недостатком для пользователя является то, что он полностью зависит от администратора хост-ЭВМ. Пользователь не может настроить рабочую среду под свои потребности – все используемое программное обеспечение является коллективным. Классическое представление централизованной архитектуры показано на рис.1.



Рис. 1. Классическое представление централизованной архитектуры Центральная ЭВМ должна иметь большую память и высокую производительность, чтобы обеспечивать комфортную работу большого числа пользователей.

Все приложения, работающие в такой архитектуре, полностью находятся в основной памяти хост-ЭВМ. Терминалы являются лишь устройствами ввода-вывода и таким образом в минимальной степени поддерживают интерфейс пользователя.

2. Архитектура "файл-сервер"

Файл-серверные приложения – приложения, схожие по своей структуре с локальными приложениями и использующие сетевой ресурс для хранения программы и данных.

- Функции сервера: хранения данных и кода программы.
 - Функции клиента: обработка данных происходит исключительно на стороне клиента. Классическое представление информационной системы в архитектуре "файл-сервер" представлено на рис. 2.

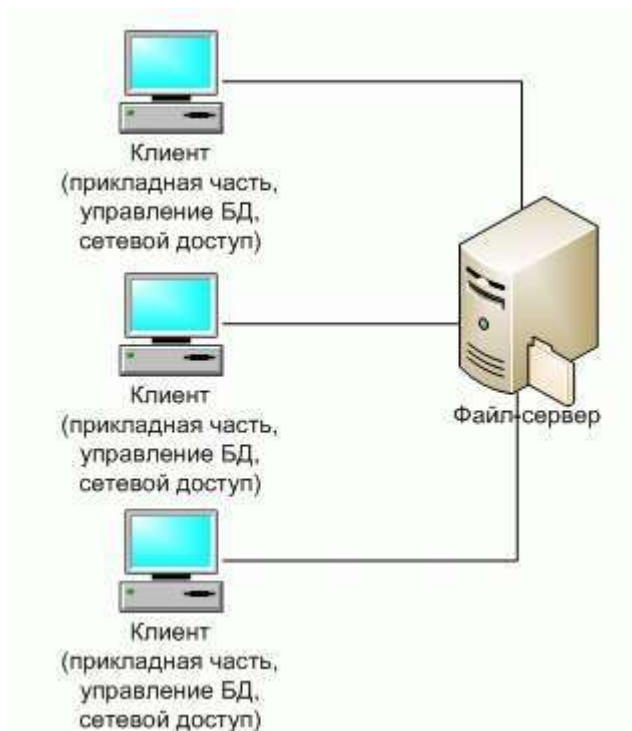


Рис. 2. Классическое представление архитектуры "файл-сервер"

Организация информационных систем на основе использования выделенных файл-серверов все еще является распространенной в связи с наличием большого количества персональных компьютеров разного уровня развитости и сравнительной дешевизны связывания РС в локальные сети.

Конечно, основным достоинством данной архитектуры является простота организации. Проектировщики и разработчики информационной системы находятся в привычных и комфортных условиях IBM PC в среде MS-DOS, Windows или какого-либо облегченного варианта WindowsServer. Имеются удобные и развитые средства разработки графического пользовательского интерфейса, простые в использовании средства разработки систем баз данных и/или СУБД.

Достоинства такой архитектуры:

- многопользовательский режим работы с данными;
- удобство централизованного управления доступом;
- низкая стоимость разработки;
- высокая скорость разработки;
 - невысокая стоимость обновления и изменения ПО.
- проблемы многопользовательской работы с данными: последовательный доступ, отсутствие гарантии целостности;
- низкая производительность (зависит от производительности сети, сервера, клиента);
- плохая возможность подключения новых клиентов;
- ненадежность системы.

Простое, работающее с небольшими объемами информации и рассчитанное на применение в однопользовательском режиме, файл-серверное приложение можно спроектировать, разработать и отладить очень быстро. Очень часто для небольшой компании для ведения, например, кадрового учета достаточно иметь изолированную систему, работающую на отдельно стоящем РС. Однако, в уже ненамного более сложных случаях (например, при организации информационной системы поддержки проекта, выполняемого группой) файл-серверные архитектуры становятся недостаточными.

5.1.4. Архитектура "клиент-сервер"

Клиент-сервер (Client-server) – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и

заказчиками услуг, называемых клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением.

Первоначально системы такого уровня базировались на классической двухуровневой клиент-серверной архитектуре (Two-tier architecture). Под клиент-серверным приложением в этом случае понимается информационная система, основанная на использовании серверов баз данных.

Схематически такую архитектуру можно представить, как показано на рис. 3.



Рис. 3. Классическое представление архитектуры "клиент-сервер"

На стороне клиента выполняется код приложения, в который обязательно входят компоненты, поддерживающие интерфейс с конечным пользователем, производящие отчеты, выполняющие другие специфичные для приложения функции.

Клиентская часть приложения взаимодействует с клиентской частью программного обеспечения управления базами данных, которая, фактически, является индивидуальным представителем СУБД для приложения.

Заметим, что интерфейс между клиентской частью приложения и клиентской частью сервера баз данных, как правило, основан на использовании языка SQL. Поэтому такие функции, как, например, предварительная обработка форм, предназначенных для запросов к базе данных, или формирование результирующих отчетов выполняются в коде приложения.

Наконец, клиентская часть сервера баз данных, используя средства сетевого доступа, обращается к серверу баз данных, передавая ему текст оператора языка SQL.

Посмотрим теперь, что же происходит на стороне сервера баз данных. В продуктах практически всех компаний сервер получает от клиента текст оператора на языке SQL.

Сервер производит компиляцию полученного оператора.

- Далее (если компиляция завершилась успешно) происходит выполнение оператора.

Преимуществами данной архитектуры являются:

- возможность, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети;
- все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов, а также на сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа;

- поддержка многопользовательской работы;

- гарантия целостности данных.

- Недостатки:
 - неработоспособность сервера может сделать неработоспособной всю вычислительную сеть;

- администрирование данной системы требует квалифицированного профессионала;

- высокая стоимость оборудования;
- бизнес логика приложений осталась в клиентском ПО.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 1. Основы проектирования информационных систем

Практическое занятие № 5 «Обоснование выбора средств проектирования информационной системы»

Цель: получение навыков построения модели управления качеством

Выполнив работу, Вы будете:

уметь:

- У3. осуществлять выбор модели и средства построения информационной системы и программных средств;
- У4. использовать алгоритмы обработки информации для различных приложений.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

1. Проанализировать методические рекомендации и выписать стандарты качества автоматизированных информационных систем.
2. Проанализировать методические рекомендации и выписать стандарты качества автоматизированных информационных систем.
3. Самостоятельно описать стандарты группы ISO, используя интернет источники.

Порядок выполнения работы:

По своему назначению и ориентации на определенные задачи и группы пользователей, документацию на ПО можно разделить на три типа:

- технологическую документацию, подготавливаемую для специалистов, ведущих разработку, сопровождение и перенос ПО на иные платформы, обеспечивающую возможность детального освоения, развития и корректировки ими программ и данных на всем жизненном цикле;
- эксплуатационную документацию, создаваемую для конечных пользователей ПО и позволяющую им осваивать и квалифицированно применять эти средства для решения конкретных функциональных задач;
- исследовательскую документацию, предназначенную для анализа характеристик,

эффективности и качества технологий и объектов проектирования с целью совершенствования методов и средств автоматизации разработки, сопровождения и переноса ПО.

Технологическая документация непосредственно и в наибольшей степени определяет процессы ЖЦ и эффективность переноса прикладных программ и данных на иные аппаратные и операционные платформы. Стандарты на эту документацию регламентируют минимальные требования к документам, сопровождающим весь жизненный цикл ПО. В ней отражаются стандарты и руководства, регламентирующие процессы разработки и обеспечения качества, требования к формализации функций, к показателям качества ПО и его компонентов, методы и средства их достижения, реальные значения достигнутых показателей качества. Реальные ограничения ресурсов, используемых в процессе разработки, квалификация специалистов, изменения внешней среды и требований заказчика объективно приводят к отклонениям реализации плана разработки от предполагавшегося. Для контроля таких изменений целесообразно предусмотреть и согласовать с заказчиком специальный документ, регламентирующий правила корректировки плана разработки ПО, а также состав и содержание поддерживающей его документации.

Ход работы:

Стандарты, регламентирующие документирование

Создание и применение АС сопровождается документированием обеспечения интерфейса с пользователями, а также возможности их освоения и функционального совершенствования. Комплексы отечественных стандартов, регламентирующие документирование АС на различных стадиях его создания, представляют во многом морально устаревшие стандарты серий «Информационная технология», «Единая система стандартов автоматизированной системы управления», «Единая система программной документации» [5]:

- ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
- ГОСТ 34.201-89 Информационная технология, комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначения документов при создании автоматизированных систем;
- РД 50-34.698-90 Информационная технология. Комплекс стандартов и руководящих указаний на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов;
- ГОСТ 24.101-80 Система технической документации на АСУ. Виды и комплектность документов;
- ГОСТ 24.102-80 Система технической документации на АСУ. Обозначение документов;
- ГОСТ 24.103-84 Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Основные положения;
- ГОСТ 24.202-80 Система технической документации на АСУ. Требования к содержанию документа «Технико-экономическое обоснование создания АСУ»;
- ГОСТ 24.203-80 Система технической документации на АСУ. Требования к содержанию общесистемных документов;
- ГОСТ 24.204-80 Система технической документации на АСУ. Требования к содержанию документа «Описание постановки задачи»;
- ГОСТ 24.205-80 Система технической документации на АСУ. Требования к содержанию документов по информационному обеспечению;
- ГОСТ 24.206-80 Система технической документации на АСУ. Требования к содержанию документов по техническому обеспечению;
- ГОСТ 24.207-80 Система технической документации на АСУ. Требования к содержанию документов по программному обеспечению;
- ГОСТ 24.208-80 Система технической документации на АСУ. Требования к

содержанию документов стадии «Ввод в эксплуатацию»;

- ГОСТ 24.209-80 Система технической документации на АСУ. Требования к содержанию документов по организационному обеспечению;
- ГОСТ 24.210-82 Система технической документации на АСУ. Требования к содержанию документов по функциональной части;
- ГОСТ 24.211-82 Система технической документации на АСУ. Требования к содержанию документа «Описание алгоритма»;
- ГОСТ 24.301-80 Система технической документации на АСУ. Общие требования к выполнению текстовых документов;
- ГОСТ 24.302-80 Система технической документации на АСУ. Общие требования к выполнению схем;
- ГОСТ 24.303-80 Система технической документации на АСУ. Обозначения условные графические технических средств;
- ГОСТ 24.304-82 Система технической документации на АСУ. Требования к выполнению чертежей;
- ГОСТ 24.401-80 Система технической документации на АСУ. Внесение изменений;
- ГОСТ 24.401-80 Система технической документации на АСУ. Учет, хранение и обращение.

В свою очередь, международные стандарты регламентируют главным образом, документирование ПО. К ним относятся:

- ISO 6592:1986. ОИ. Руководство по документации для вычислительных систем;
- ISO 9294:1990-ТО. ИТ. Руководство по управлению документированием программного обеспечения;
- ISO 9127. ИТ. Пользовательская и рекламная документация на пакеты программ.

Поскольку ГОСТы доступны широкому кругу отечественных специалистов, ниже более детально рассматриваются требования международных стандартов.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 1. Основы проектирования информационных систем

Практическое занятие № 6 «Описание бизнес-процессов заданной предметной области»

Цель: произвести описание бизнес-процессов

Выполнив работу, Вы будете:

уметь:

- У1. осуществлять постановку задач по обработке информации;
- У2. проводить анализ предметной области;
- У3. осуществлять выбор модели и средства построения информационной системы и программных средств;
- У4. использовать алгоритмы обработки информации для различных приложений.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства.

Задание:

1. По выбранному варианту (практическая работа 1) составить функциональную схему программного обеспечения.
2. По составленной функциональной схеме (задание 1) произвести процесс декомпозиции функциональной схемы. Примерное количество блоков должно составлять не менее 3-4.

Порядок выполнения работы

Входы одного блока могут быть выходами или управлением для других.



Функциональная модель бизнес-процесса расчета и оптимизации себестоимости ремонта и модернизации асинхронного двигателя

Функциональная модель начинается с построения общего описания процесса - контекстной диаграммы (диаграмма нулевого уровня). На этом уровне весь процесс рассматривается как один функциональный блок со всеми связанными обрабатываемыми и управляющими объектами.

Затем контекстная диаграмма детализируется на другой диаграмме с помощью нескольких блоков, соединенных между собой интерфейсными дугами. Они определяют основные функции исходного бизнес-процесса. Данная декомпозиция выявляет полный набор подфункций, каждая из которых показана как блок.

Декомпозиция процесса «Расчета и оптимизации себестоимости ремонта и модернизации асинхронного двигателя» может быть представлена следующими задачами:

- расчет параметров асинхронного двигателя и его комплектующих: номинальные данные ремонтируемого двигателя (номинальная частота вращения, номинальная частота тока, номинальный $\cos\varphi$ $\cos\varphi = 1,0$ $\cos\varphi = 1,0$, номинальный КПД, номинальное фазное

напряжение); размеры магнитной системы электрической машины (число обмоток, фаз, пар полюсов, пазов статора и ротора, длина статора, размер паза); тип обмотки («звезда» или «треугольник»); вид ротора (фазный или короткозамкнутый);

- процесс расчета себестоимости ремонта и модернизации синхронного двигателя: расчет стоимости трудовых и материальных затрат с учетом квалификации специалистов, выполняющих технологические операции, цены рабочего времени специалиста, длительности выполнения операции, а также особенности расчета параметров и изготовления обмотки статора асинхронного двигателя, стоимость материалов. Помимо прямых затрат вводятся коэффициенты, учитывающие число и форму пазов статора, число слоев обмотки статора, длину магнитной системы статора, интеллектуальную и инновационную составляющие;

- оптимизация процессе ремонта и модернизации асинхронного двигателя. Под оптимизацией процесса понимается понижение затрат на ремонт и модернизацию за счет распараллеливания технологических операций. Оптимизация предусматривает два варианта:

- вариант 1 – минимизация трудовых и материальных затрат для заданного времени ремонта - так называемого, «срочный ремонт»;

- оптимизация по варианту 2 – минимизация трудовых и материальных затрат при среднестатистическом времени ремонта.

- формирование сметы с указанием специалиста, назначенного на определенную операцию, стоимости технологической операции, стоимости и количество материального ресурса, выгружается в базу данных. Также в модуле составления отчета представляются графики затрат по трудовым и материальным ресурсам. Помимо этого, в данном модуле строится сетевая модель ремонта и модернизации.

На рисунке 2 изображена декомпозиция бизнес-процесса «Расчета и оптимизации себестоимости ремонта и модернизации асинхронного двигателя», построенная с использованием программного продукта MS Visio.

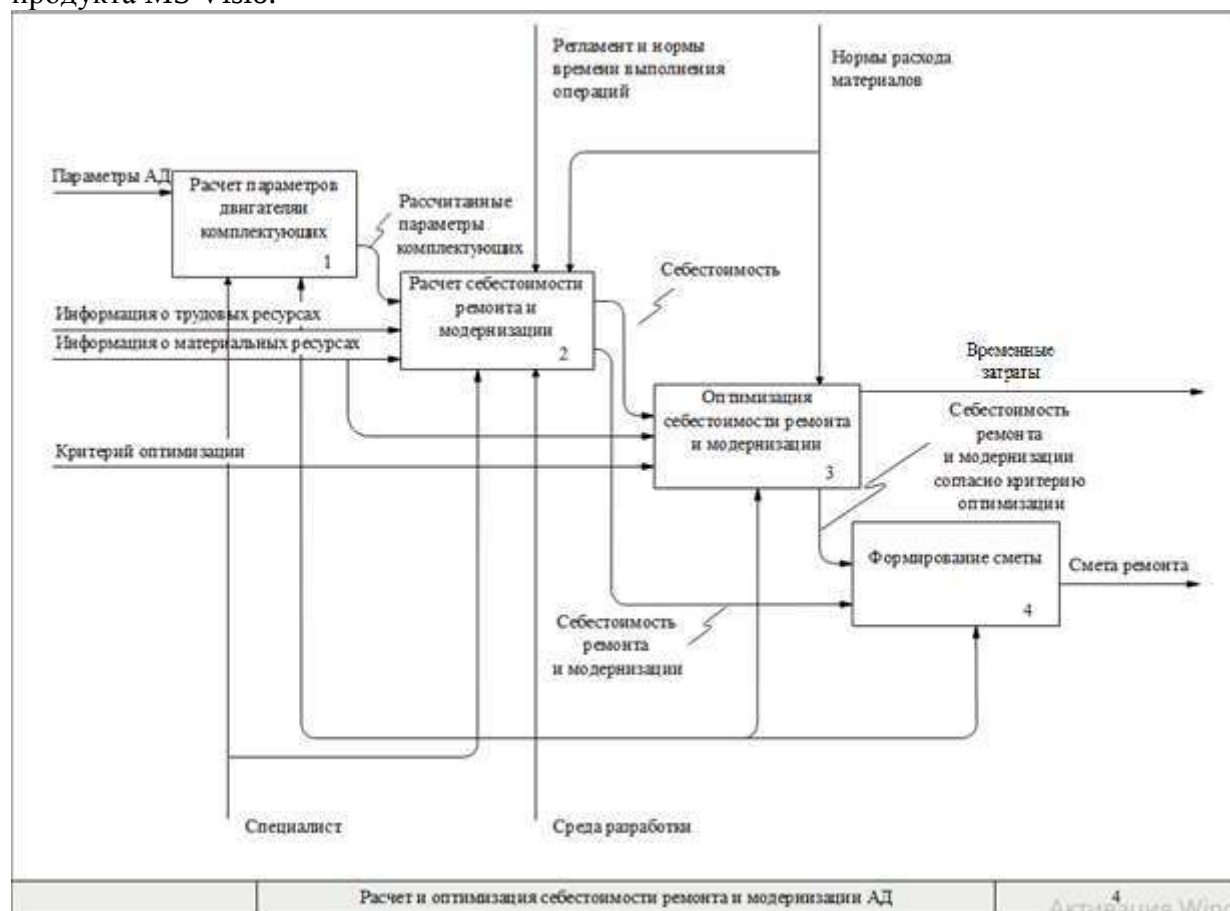


Рисунок 2 - Детализация функциональной схемы программного обеспечения

Декомпозиция каждой функции может осуществляться до тех пор, пока на диаграмме не будут отображены элементарные операции. Число уровней декомпозиции не ограничено и определяется целями моделирования. Обычно для структурного анализа бизнес-процессов используется 2-3 уровня декомпозиции. Последующие уровни декомпозиции требуются для построения алгоритмов обработки информационных потоков при проектировании и разработке информационных систем, а также для разработки инструкций для исполнителей бизнес-процессов.

Ход работы:

Бизнес-процесс - целенаправленная деятельность по преобразованию исходных материалов и информации в конечный продукт, услугу, информацию в соответствии с предварительно установленными правилами.

Функция (или процедура) - упорядоченная последовательность операций, предназначенная для получения промежуточного результата бизнес-процесса.

Операция - ряд упорядоченных действий, рассматривать которые в отдельности в рамках создаваемой модели нецелесообразно [6].

В рамках этапа формирования требований к информационной системе происходит процесс моделирования бизнес-процессов организации, где отражаются существующие потоки информации, работ, деятельности в виде модели процессов. При моделировании бизнес-процесс происходит детальное описание компонентов.

Распространенными методами построения моделей бизнес-процессов являются структурное и объектно-ориентированное моделирование.

Структурный подход к моделированию бизнес-процессов заключается в представлении бизнес-процессов в виде последовательности функций с декомпозицией до неделимых операций [6].

Достоинством структурного подхода является графическая простота и наглядность представления бизнес-процессов. Существенным недостатком структурного подхода является некоторая субъективность детализации операций и, как следствие, большая трудоемкость в построении моделей.

Объектно-ориентированный подход основан на представлении предметной области в виде объектов, взаимодействующих между собой посредством передачи сообщений. Основным понятием объектно-ориентированного подхода является объект. Каждый объект характеризуется некоторым набором присущих ему свойств и операций, выполняемых этим объектом. Для описания бизнес-процессов с использованием объектно-ориентированной методологии в настоящее время широко применяется унифицированный язык моделирования UML (Unified Modeling Language).

В лабораторном занятии 6 рассматривается моделирование бизнес-процессов на основе структурного подхода на примере построения IDEF0 (Integration computer aided manufacturing Definition).

Стандарт IDEF0 представляет организацию как набор модулей, здесь существует правило — наиболее важная функция находится в верхнем левом углу, кроме того есть правило стороны:

- стрелка входа всегда приходит в левую кромку активности,
- стрелка управления — в верхнюю кромку,
- стрелка механизма — нижняя кромка,
- стрелка выхода — правая кромка.

Описание выглядит как «чёрный ящик» с входами, выходами, управлением и механизмом, который постепенно детализируется до необходимого уровня. Также для того чтобы быть правильно понятым, существуют словари описания активностей и стрелок. В этих словарях можно дать описания того, какой смысл вы вкладываете в данную активность либо стрелку. IDEF0 используется два графических элемента:

- функциональный блок - описание функций, операций, действий;
- интерфейсная дуга, связывающая два функциональных блока. Каждый блок имеет входы четырех типов, рисунок 1:
- вход (входная информация);
- выход (выходная информация);

- управление (управляющая информация);
- механизм (исполнитель, который осуществляет операцию; информационная система и пр.).

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 2. Система обеспечения качества информационных систем

Практическое занятие № 1

Построение модели управления качеством процесса изучения модуля «Проектирование и разработка информационных систем»

Цель: получение навыков построения модели управления качеством

Выполнив работу, Вы будете:

уметь:

- УЗ. осуществлять выбор модели и средства построения информационной системы и программных средств.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства.

Задание:

1. Проанализировать имеющиеся стандарты обеспечения качества информационной системы.

Порядок выполнения работы:

1. Привести национальные стандарты обеспечения качества автоматизированных информационных систем.
2. Охарактеризовать международную систему стандартизации и сертификации качества продукции.
3. Описать стандарты группы ISO.
4. Привести методы контроля качества в информационных системах.
5. Постройте модель управления качеством процесса изучения модуля «Проектирование и разработка информационных систем».

Ход работы:

Стандарты, регламентирующие документирование

Создание и применение АС сопровождается документированием обеспечения интерфейса с пользователями, а также возможности их освоения и функционального совершенствования.

Комплексы отечественных стандартов, регламентирующие документирование АС на различных стадиях его создания, представляют во многом морально устаревшие стандарты серий «Информационная технология», «Единая система стандартов автоматизированной системы управления», «Единая система программной документации» [5]:

- ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
- ГОСТ 34.201-89 Информационная технология, комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначения документов при создании автоматизированных систем;
- РД 50-34.698-90 Информационная технология. Комплекс стандартов и руководящих указаний на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов;
- ГОСТ 24.101-80 Система технической документации на АСУ. Виды и комплектность документов;
- ГОСТ 24.102-80 Система технической документации на АСУ. Обозначение документов;
- ГОСТ 24.103-84 Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Основные положения;
- ГОСТ 24.202-80 Система технической документации на АСУ. Требования к содержанию документа «Технико-экономическое обоснование создания АСУ»;
- ГОСТ 24.203-80 Система технической документации на АСУ. Требования к содержанию общесистемных документов;
- ГОСТ 24.204-80 Система технической документации на АСУ. Требования к содержанию документа «Описание постановки задачи»;
- ГОСТ 24.205-80 Система технической документации на АСУ. Требования к содержанию документов по информационному обеспечению;
- ГОСТ 24.206-80 Система технической документации на АСУ. Требования к содержанию документов по техническому обеспечению;
- ГОСТ 24.207-80 Система технической документации на АСУ. Требования к содержанию документов по программному обеспечению;
- ГОСТ 24.208-80 Система технической документации на АСУ. Требования к содержанию документов стадии «Ввод в эксплуатацию»;
- ГОСТ 24.209-80 Система технической документации на АСУ. Требования к содержанию документов по организационному обеспечению;
- ГОСТ 24.210-82 Система технической документации на АСУ. Требования к содержанию документов по функциональной части;
- ГОСТ 24.211-82 Система технической документации на АСУ. Требования к содержанию документа «Описание алгоритма»;
- ГОСТ 24.301-80 Система технической документации на АСУ. Общие требования к выполнению текстовых документов;
- ГОСТ 24.302-80 Система технической документации на АСУ. Общие требования к выполнению схем;
- ГОСТ 24.303-80 Система технической документации на АСУ. Обозначения условные графические технических средств;
- ГОСТ 24.304-82 Система технической документации на АСУ. Требования к выполнению чертежей;
- ГОСТ 24.401-80 Система технической документации на АСУ. Внесение изменений;
- ГОСТ 24.401-80 Система технической документации на АСУ. Учет, хранение и обращение.

В свою очередь, международные стандарты регламентируют главным образом, документирование ПО. К ним относятся:

- ISO 6592:1986. ОИ. Руководство по документации для вычислительных систем;

- ISO 9294:1990-ТО. ИТ. Руководство по управлению документированием программного обеспечения;
- ISO 9127. ИТ. Пользовательская и рекламная документация на пакеты программ.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 2. Система обеспечения качества информационных систем Практическое занятие № 2 Реинжиниринг методом интеграции

Цель: произвести реинжиниринг методом интеграции

Выполнив работу, Вы будете:

уметь:

- У2. проводить анализ предметной области;
- У3. осуществлять выбор модели и средства построения информационной системы и программных средств;
- У4. использовать алгоритмы обработки информации для различных приложений.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание

1. Проанализировать процесс реинжиниринга своей предметной области (практическая работа1). Выявить основные этапы реинжиниринга.

Порядок выполнения работы:

1. Ознакомьтесь с краткими теоретическими сведениями
2. Выполните задание
3. Представьте выполненную работу в виде отчета

Ход работы:

Реинжиниринг в качестве приема инновационного менеджмента затрагивает инновационный процесс, направленный как на производство новых продуктов и операций, так и на их реализацию, продвижение, диффузию. Поскольку конечной целью реинжиниринга являются нововведения (т.е. инновации), то реинжиниринг в более узком понимании есть реинжиниринг инноваций.

Реинжиниринг представляет собой инженерно-консультационные услуги по перестройке предпринимательской деятельности на основе производства и реализации инноваций. Американский ученый М. Хамлер, который ввел в научный оборот термин «реинжиниринг», дал ему такое определение.

Реинжиниринг — это фундаментальное переосмысление и радикальное перепроектирование деловых процессов для достижения резких, скачкообразных улучшений современных показателей деятельности компаний, таких как стоимость, качество, сервис и темпы.

Это определение содержит четыре ключевых понятия: «фундаментальный», «радикальный», «резкий» («скачкообразный») и «процесс». Хотя понятие «процесс» («бизнес-процесс») является в данном контексте наиболее важным, его трудно объяснить менеджерам, так как большинство из них привыкло иметь дело с задачами, работами, структурами, людьми, но не с процессами.

Основные этапы реинжиниринга

Бизнес-процесс - это система последовательных, целенаправленных и регламентированных видов деятельности, в которой посредством управляющего воздействия и с помощью ресурсов входы процесса преобразуются в выходы, т.е. результаты процесса, представляющие ценность для потребителей (рис. 1).



Рис. 1. Типичные бизнес-процессы, проектируемые и совершенствуемые в ходе реинжиниринговой деятельности

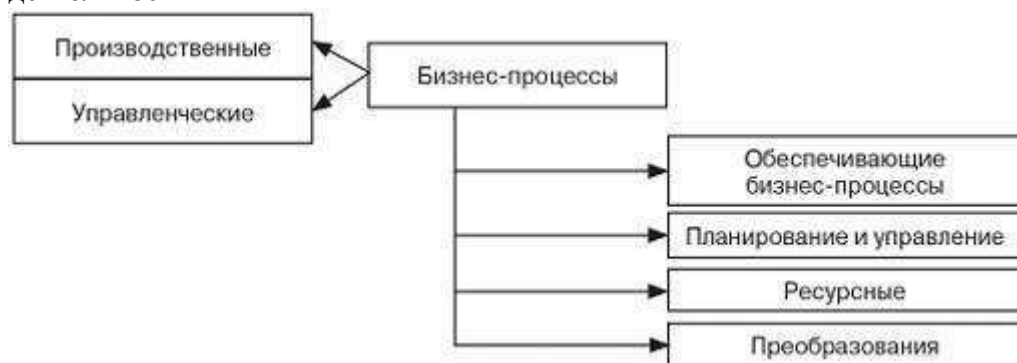


Рис. 2. Классификация бизнес-процессов

Эффективность бизнес-процессов зависит от ряда показателей (рис. 3).



Рис. 3. Показатели эффективности бизнес-процессов

Направления улучшения управления бизнес-процессами приведены на рис. 4.



Рис. 4. Улучшение управления бизнес-процессами

Реинжиниринг - это фундаментальное переосмысление и радикальное перепроектирование организации и ее важнейших процессов.

Основопологающей целью реинжиниринга бизнес-процессов является гибкое и оперативное приспособление к ожидаемым изменениям запросов потребителей: соответствующее изменение стратегии, технологии, организации производства и управления на основе эффективной компьютеризации.

Результатом реинжиниринга является улучшение важнейших количественно измеряемых показателей.

Реинжиниринг бывает успешным, только если он находится в тесной связи с организационным развитием организации.

По степени воздействия на организационную структуру различают эволюционный и революционный реинжиниринг бизнес-процессов.

При **эволюционном** реинжиниринге оптимизируется внутренняя интеграция различных бизнес-процессов, но не вносятся существенные изменения в функционирование организации.

При **революционном** реинжиниринге перепроектируются все бизнес-процессы и происходит переориентация организации на новый вид бизнеса.

При реинжиниринге осуществляется либо перепроектирование, либо совершенствование бизнес-процессов (табл. 1).

Таблица 1. Сущность реинжиниринга

Вид реинжиниринга	Ситуация для применения
Кризисный реинжиниринг (перепроектирование бизнес-процессов)	Состояние глубокого кризиса (потеря конкурентоспособности, отказ потребителей от товаров и т.п.)

Реинжиниринг (совершенствование процессов)	развития бизнес-	Удовлетворительное текущее положение при нежелательных тенденциях и неблагоприятных прогнозах. Благополучная ситуация при желании ускорить и увеличить отрыв от конкурентов
--	------------------	---

В общем виде реинжиниринг решает следующие задачи:

- способствует созданию сети связей для чрезвычайных условий (поскольку развивает горизонтальные управленческие связи);
- создает организационные предпосылки для централизации информационных потоков (поскольку способствует получению информации, систематизированной по конкретным процессам);
- содействует разделению функций высшего руководства и созданию сети оперативных групп (поскольку позволяет применить для этих целей технологию работы процессных команд);
- мотивирует творческий подход, анализирует ситуации и коллективную работу (поскольку на основе этих принципов видоизменяются характер работы и роль работников при реинжиниринге);
- позволяет успешно совмещать координацию стратегии из центра и децентрализованное исполнение решений (поскольку опирается на смешанные процессы и матричные структуры управления);
- создает организационные условия реструктуризации предприятия (поскольку увязывает изменения структуры управления с деятельностью процессных команд).

Базовые категории реинжиниринга отражены в табл. 2. Таблица 2. Базовые категории реинжиниринга

Базовые категории	Сущность категории
Деловая процедура	Функция, задача, цепь событий, происходящих в течение определенного времени и обладающих распознаваемым результатом
Бизнес-процесс	Горизонтальная иерархия внутренних и зависимых между собой функциональных действий, конечной целью которых является выпуск продукции или отдельных
Бизнес-система	Связанное множество бизнес-процессов, конечной целью которого является выпуск продукции в виде товаров, услуг, документов

К основным **принципам** реинжиниринга бизнес-процессов относятся:

- концентрация ответственности: несколько рабочих процедур объединяются в одну (горизонтальное сжатие процесса);
- делегирование полномочий в сочетании с самоконтролем: исполнители принимают самостоятельные решения (вертикальное сжатие процесса);

- естественный порядок выполнения процесса, например параллельный, а не последовательный;
 - различные варианты исполнения процессов в зависимости от ситуации;
 - выполнение работы там, где ее можно сделать наиболее эффективно (вплоть до ее передачи клиенту процесса);
 - сокращение объема согласований, проверок и контроля — контроль со стороны менеджеров заменяется по возможности контролем со стороны потребителей данного процесса.
- Возможные характеристики процесса реинжиниринга по основным критериям отражены в табл. 3.

3.
Таблица 3. Характеристика реинжиниринга по основным критериям

Критерии	Реинжиниринг
Происхождение метода	Инженерные науки, консультационная практика по менеджменту
Основная идея	Радикальное переосмысление и перепроектирование предприятия или производственно-хозяйственных процессов
Принципиальная позиция менеджера	Мышление дискретными категориями, четкая постановка вопроса «Почему нужны перемены?», привлечение к участию убежденных сторонников перемен
Характер изменений	Глубокие и всеохватывающие перемены, прерывность процесса, скачкообразные изменения
Сроки реализации проектов	Несколько лет с упором на быстрый успех, измеряемый в количественных показателях
Объект изменений	Предприятие в целом или ключевые процессы
Цели	Значительное и стабильное повышение рентабельности (экономической эффективности)
Вид кризиса	Кризис ликвидности, кризис успеха
Стратегия изменений	Стратегия «сверху вниз»
Ключевые роли	Лидер, группа реинжиниринга, специалисты
Методические аспекты	Реорганизация ключевых процессов в

	соответствии с принятой рыночной стратегией, адаптация организационных структур и должностных инструкций, изменение ценностных представлений,
	внедрение современной информационной технологии, развитие персонала и новые методы оплаты труда
Сильные стороны	Возможность радикального обновления, шансы явного повышения рентабельности, быстрота изменений, концептуальное единство мероприятий, значительное расширение компетенции специалистов
Слабые стороны	Нестабильность в фазе изменений, ограничения во времени и действиях в связи с желанием быстро улучшить результаты, низкая социальная приемлемость

Особое внимание в методике реинжиниринга уделяется формированию команды и ее взаимодействию с коллективом организации. Процессные команды замещают старую структуру управления.

В зависимости от выполняемых работ наиболее часто используются два типа процессных команд:

- команда объединяет совместно работающих людей разных специальностей, выполняющих рутинную и повторяющуюся работу, на длительное время;
- команда объединяет людей для решения нестандартной и, как правило, сложной задачи. Команды подобного типа создаются на время решения задачи и по завершении проекта расформируются. Причем один работник может быть одновременно членом нескольких команд, распределяя свое время между несколькими проектами.

Определение **участников** реинжиниринговой деятельности и выполняемых ими функций являются основой эффективности процесса (табл. 4).

Таблица 4. Участники реинжиниринговой деятельности и их функции

Участники	Функции
1. Лидер проекта — один из высших менеджеров фирмы	Возглавляет деятельность реинжиниринга, отвечает за идеологическое обоснование проекта, создает общий дух новаторства и ответственности

2. Управляющий комитет — члены высшего руководства, лидер проекта, менеджеры процессов	Осуществляет наблюдение, согласует цели и стратегии, интересы рабочих команд, разрешает конфликты
3. Менеджеры оперативного руководства	Разрабатывают методики и инструменты реинжиниринга, проводят обучение, координируют, помогают в формировании команд
4. Менеджеры процессов	Разрабатывают методики и инструменты реинжиниринга, проводят обучение, координируют, помогают в формировании команд
5. Рабочие команды — работники фирмы и внешние консультанты и разработчики	Осуществляют непосредственную работу по реинжинирингу

Для формирования слаженной работоспособной команды необходимы:

- точное описание основных целей;
- тщательная разработка бюджета;
- идентификация ключевых ролей и фиксация объективных требований к кандидатам;
- тщательный подбор и детальная проверка кандидатов;
- непрерывный мониторинг, умение выявлять и исправлять ошибки.

Для осуществления реинжиниринга используются определенные **инструменты** (рис. 5).

Этапы проведения реинжиниринга приведены на рис. 6.

В общем случае этапы методики реинжиниринга бизнес-процессов включают следующие действия:

- разработка проекта и выделение бизнес-процессов. Определяются цели и задачи проекта, формируется команда по реинжинирингу и определяется подход к реинжинирингу;
- документирование бизнес-процессов. На этом этапе выполняется построение графических моделей бизнес-процессов на основе предложенной методики их документирования, хронометрируются составные операции бизнес-процессов;
- сравнительный анализ бизнес-процессов (бенчмаркинг). Проводится анализ бизнес-процессов с целью сравнения их с бизнес- процессами передовых подразделений организации или организации-конкурента;
- разработка образа будущей организации. Цель этого этапа- сформулировать систему взглядов на новую организацию согласно ее целям и возможностям. В команду по реинжинирингу целесообразно включать субъектов внешней среды;
- анализ проблем и перепроектирование бизнес-процессов и технологий. Цель этапа — выявление проблемных мест технологических и бизнес-процессов;
- внедрение новых бизнес-процессов, технологий и оценка результатов. На этом этапе необходимо сопоставить результаты эффективности функционирования бизнес-процессов с заданными в начале реинжиниринга критериями, с учетом затрат по видам функциональной

деятельности.



Рис. 5. Инструменты реинжиниринга

Успех процесса реинжиниринга обусловлен определенными факторами (рис. 7). Реинжиниринг в организациях приобретает ярко выраженную антикризисную направленность. Направленность организационных мероприятий при антикризисном управлении совпадает с динамикой изменений, происходящих при реинжиниринге.

При этом необходимо отметить следующее:

- эффективная деятельность процессных команд может обусловить создание новых структурных единиц;
- создание или объединение процессных команд может обусловить дробление или укрупнение предприятия;
- создание процессных команд в новых видах деятельности способствует переходу к дивизиональной модели управления предприятием.

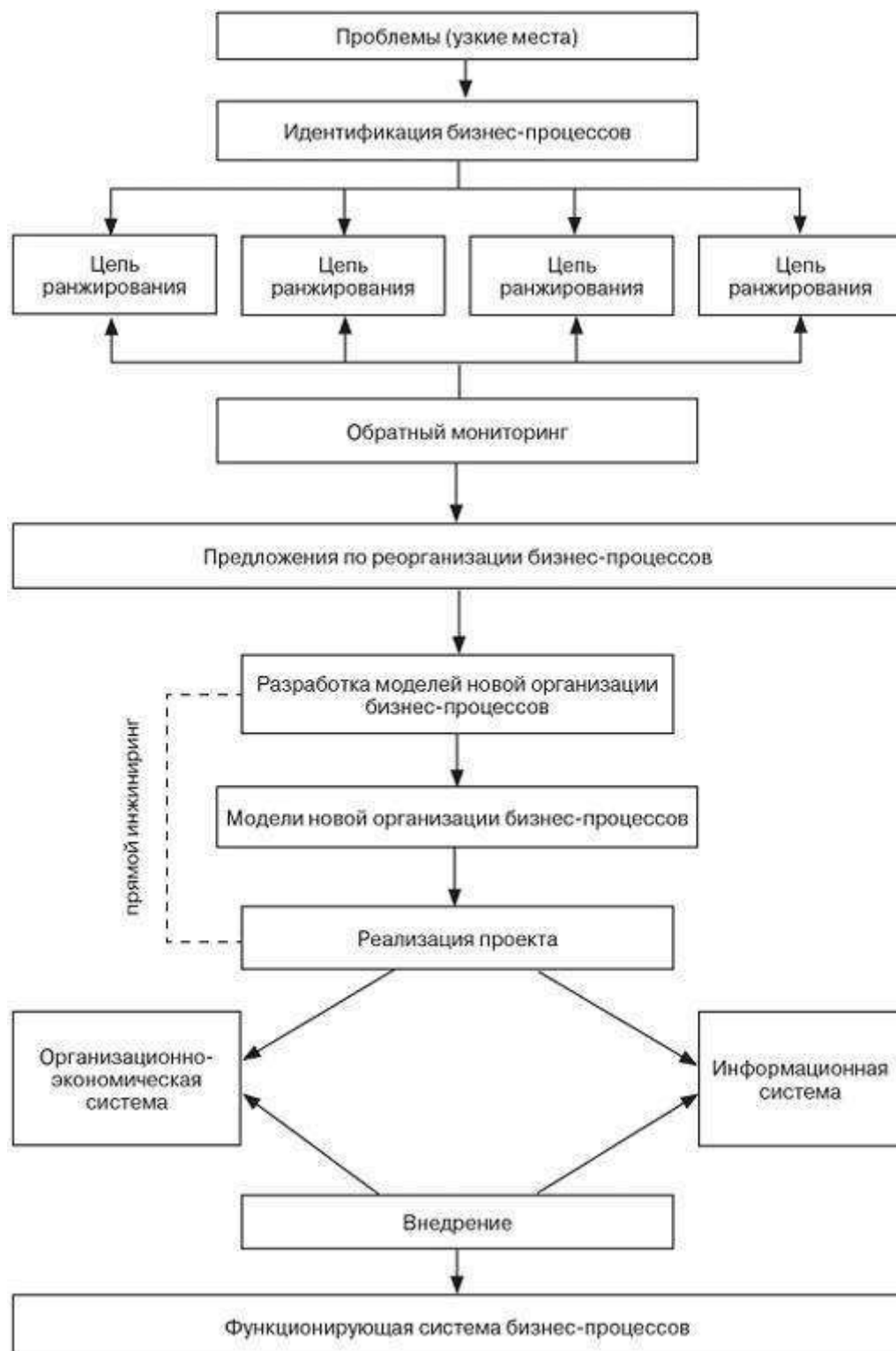


Рис. 6. Этапы реинжиниринга бизнес-процессов



Рис. 7. Факторы успеха реинжиниринга

Тем самым реинжиниринг повышает мобильность организации при переходе на антикризисные технологии управления, что особенно важно в современных российских условиях.

Реинжиниринг бизнес-процесса (РБП) — фундаментальное переосмысление и радикальная модификация бизнес-процессов для достижения перелома в работе по совершенствованию в критических текущих показателях, таких как затраты, качество, обслуживание и скорость. РБП — это философия совершенствования.

Задача РБП — достижение фундаментальных улучшений путем перепроектирования процесса таким образом, чтобы максимизировать добавление ценности, а прочие показатели минимизировать.

Этот подход можно применить как на уровне отдельного процесса, так и на уровне целой организации.

В процедуре проведения РБП можно выделить четыре фазы (рис. 8).

1. **Планирование.** Определяется проект РБП, формируется команда проекта и, если это возможно, определяются цели проекта.
2. **Реинжиниринг,** основанный на существующем процессе. Набор методов позволяет перестроить процесс, поднять его уровень, чтобы в результате резко его улучшить.
3. **Преобразование.** Определяют, как внедрить новый процесс с учетом существующего процесса, потребных инвестиций, обучения и т.п.
4. **Внедрение.** Решения, выработанные и утвержденные на двух предыдущих фазах, внедряются и меняется процесс.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 2. Система обеспечения качества информационных систем

Практическое занятие № 3

Разработка требований безопасности информационной системы

Цель: получение навыков разработки требований безопасности информационной системы

Выполнив работу, Вы будете:

уметь:

- У14. использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Порядок выполнения работы

1. Определите цели и задачи системы защиты информации.
2. Перечислите факторы, влияющие на организацию системы защиты информации.
3. Определите дестабилизирующие воздействия на информационную систему и способы их нейтрализации.
4. Напишите программу по подсчету общей вероятности нарушения безопасности

$$P = \sum_{j=1}^k \sum_{i=1}^n P_i p(j/i) q_{H1} (q_{H2} + [1 - \exp(-\alpha t_{от})] (1 - q_{H2}))$$

объекта, подсчитываемой по формуле

где k – число угроз; n – число нарушителей; P_i – вероятность появления субъекта i -го типа; $p(j/i)$ – условная вероятность того, что субъект i -го типа выберет для реализации угрозу j -го типа; q_{H1} – вероятность несрабатывания средств обнаружения; q_{H2} – вероятность несрабатывания средств отражения; α – постоянная величина, характеризующая "скорость" реализации угрозы, $t_{от}$ – время, которым располагает субъект угрозы, если $t_{от} = 0$ – угроза не реализуется.

5. Разработайте требования безопасности информационной системы (см. практическая работа № 1). Выберите методы и средства защиты информации для исследуемой информационной системы.

Ход работы:

Информационная безопасность – обеспечение конфиденциальности и целостности информации, недопущение несанкционированных действий с ней, в частности, ее использования, раскрытия, искажения, изменения, исследования и уничтожения. Положения информационной безопасности одинаковы для всех форм хранения информации: физических, цифровых или любых других. С появлением компьютеризированных информационных систем безопасность данных вышла на первый план.

Основные цели информационной безопасности

Использование систем информационной безопасности ставит перед собой конкретные задачи по сохранению ключевых характеристик информации и обеспечивает:

- конфиденциальность данных – доступ есть только у лиц, имеющих на это полномочия;
- доступность информационных систем с находящимися в них данными конкретным пользователям, у которых есть право доступа к таким сведениям;
- целостность данных предполагает блокировку несанкционированного изменения информации;
- подлинность – полнота и общая точность информации;
- неотказуемость – возможность определить источник или авторство информации.

Главная цель систем информационной безопасности – гарантия защиты данных от внешних и внутренних угроз.

Для обеспечения в информационной системе полной конфиденциальности применяются четыре метода, актуальных для любого формата информации:

- ограничение или полное закрытие доступа к информации;
- шифрование;
- дробление на части и разрозненное хранение;
- скрывание самого факта существования информации.

Виды информационных угроз

Для определения правильной стратегии информационной защиты необходимо определить, что угрожает безопасности данных. Угрозы информационной безопасности – вероятные события и действия, которые в перспективе способны привести к утечке или потере данных, несанкционированному доступу к ним. Это, в свою очередь, приведет к моральному или материальному ущербу.

Первичное свойство угроз информационных систем – происхождение, по которому угрозы делятся на следующие:

Техногенные источники – угрозы, вызванные проблемами с техническим обеспечением, их прогнозирование затруднительно.

Антропогенные источники – угрозы человеческих ошибок. Могут быть как нечаянными, так и преднамеренными атаками. Непреднамеренные угрозы – это случайная ошибка, например, пользователь по незнанию отключил антивирус. Как правило, антропогенные проблемы можно спрогнозировать для принятия предупредительных мер. Умышленные угрозы безопасности приводят к информационным преступлениям.

Стихийные источники – непреодолимые обстоятельства, которые имеют низкую вероятность прогнозирования, и их предотвращение невозможно. Это различные стихийные бедствия, пожары, землетрясения, прекращение подачи электричества из-за ураганов.

Также по расположению источника относительно информационной системы угрозы классифицируются на внешние и внутренние. Особенно применительно такое разделение к масштабным системам государственного уровня.

Если внешний уровень – это атаки со стороны хакеров, компаний-конкурентов или враждебно настроенных государств, то внутренние угрозы обусловлены:

- низким уровнем программно-технического обеспечения;
- на государственном уровне – плохим развитием технологии передачи данных и ИТ-сектора в целом;
- низкой компьютерной грамотностью пользователей.

Основная цель систем информационной безопасности – ликвидация внутренних угроз. Они обычно состоят в следующем:

- происходит изъятие данных злоумышленниками, уничтожаются информационные пакеты, что нарушает работу информационной среды;
- сотрудники создают бэкдоры или сливают информацию;
- шпионские программы незаметно воздействуют на рабочий код и системное оборудование.

Поэтому фактически вся работа системы информационной безопасности сводится к созданию безопасных каналов связи, защите серверов, обеспечению безопасности внешних носителей и рабочих мест пользователей.

Информационная безопасность в компьютерной среде

Вопрос поддержания безопасности информационных систем одинаково остро стоит и перед обычными пользователями, и перед предприятиями. Потеря данных для компаний влечет за собой, прежде всего, потерю доверия и репутации. Для человека же в лучшем случае утечка выливается в навязчивый показ таргетированной рекламы, в худшем – конфиденциальная информация (пароли, данные банковских карт, сведения для входа в системы) может быть использована мошенниками в корыстных целях.

Для контроля данных, обращающихся в информационной среде, используют различные программные средства:

- сверхмощные приложения, работа которых состоит в обеспечении безопасности и шифровании финансовых и банковских реестров информации;
- глобальные решения, работающие на уровне всего информационного массива;
- утилиты для решения конкретных проблем.
-

Методы защиты информационных систем

Смысл информационной защиты в сохранении информации в первозданном виде, исключая доступ посторонних.

Системный подход состоит из четырех составляющих обеспечения безопасности:

- законы и нормативно-правовые акты;
- распределение задач между ИБ-подразделениями;
- политика информационной безопасности;
- техническое и программное обеспечение.

Всем методам защиты на предприятии присущи такие характеристики:

- применение технических средств, фактическое использование которых растет по мере расширения информационного пространства и количества рабочих станций;
- постоянный мониторинг баз данных;
- непрерывная разработка новых вычислительных систем с улучшенными методами шифрования, постоянное шифрование уже имеющимися методами;
- ограничение доступа к информации на предприятии.

Наиболее серьезную угрозу для информационных систем представляют компьютерные вирусы. Они приносят больше всего убытков информационной инфраструктуре. Основная проблема состоит в том, что антивирусное программное обеспечение не может полностью предупредить появление новых угроз. Вследствие этого, так или иначе, происходит повреждение информационных пакетов и нарушение работы информационных систем. Устранение проблемы возможно только по факту после нахождения вредоносного вмешательства. Также стоит упомянуть физические методы защиты информации – устройства, мгновенно распознающие стороннее вмешательство в систему.

Чтобы защитить определенный объект от внешних и внутренних угроз, необходимо создание Системы обеспечения информационной безопасности (СОИБ).

Для построения эффективной и работоспособной системы руководствуются примерным планом:

- выявляют необходимые данному объекту степени защиты;
- соотносят их с положениями законов и постановлений, действующих на территории страны в этом направлении деятельности;
- обращаются к предыдущим наработкам;
- назначают ответственные подразделения и распределяют ответственность между ними;
- определяют политику информационной безопасности данного объекта и используют необходимые программно-технические методы для ее внедрения.

В итоге происходит создание системы менеджмента информационной безопасности на предприятии, при помощи которой производится постоянный и действенный контроль эффективности средств защиты, своевременная коррекция предпринимаемых действий и пересмотр существующих положений для обеспечения максимальной безопасности.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 2. Система обеспечения качества информационных систем

Практическое занятие № 4

Реинжиниринг бизнес-процессов методом горизонтального и/или вертикального сжатия

Цель: произвести реинжиниринг бизнес-процессов методом горизонтального и/или вертикального сжатия

Выполнив работу, Вы будете:

уметь:

- У14. использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

1. Произвести реинжиниринг бизнес-процессов методом горизонтального и/или вертикального сжатия.

Порядок выполнения работы:

1. Ознакомьтесь с краткими теоретическими сведениями
2. Выполните задание
3. Представьте выполненную работу в виде отчета

Ход работы:

Существует два принципиально разных способа применения РБП (рис. 9).

1. **Систематический реинжиниринг** — текущий процесс понят, документирован и проан

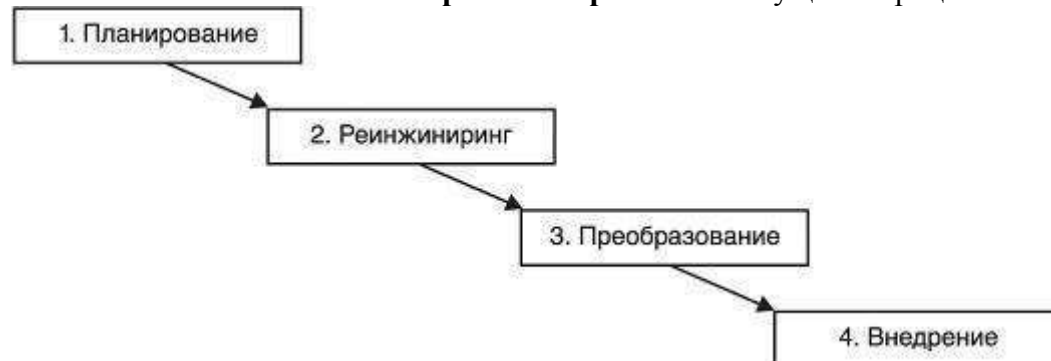


Рис. 8. Фазы процедуры проведения реинжиниринга бизнес-процесса



Рис. 9. Способы применения РБП

Этот вид реинжиниринга проводится, опираясь на правила объединить, автоматизировать.

1-й этап. Уничтожить. Исключение всех операций, которые не связаны с добавлением ценности. Эта задача очень актуальна. Например, в компании «Тойота» в расчетах большинства традиционных производственных процессов принимается, что в любое время суток 85% сотрудников заняты непродуктивной работой:

- 5% сотрудников могут наблюдать, но не действовать;
 - 25% сотрудников чего-то ждут;
 - 30% сотрудников делают что-то, что увеличивает уровень запасов, но не добавляет ценности;
 - 25% работают, однако пользуются устаревшими стандартами и процедурами.
- 2-й этап. **Упростить.** После освобождения от всего лишнего нужно упростить до максимума все, что осталось. Обычно действие «упростить» проводится на особо сложных участках.
- 3-й этап. **Объединить.** Происходит дальнейшее облегчение движения потока от поставщика к организации и от организации к потребителю. Предприятия, которые добились создания особенно тесных связей со своими поставщиками и потребителями, обычно находятся в более выгодном положении. Они имеют большие возможности для успешного совершенствования. Типично интегрировать поставщиков основных комплектующих и их потребителей в единый процесс разработки продукции.
- 4-й этап. **Автоматизировать.** Информационные технологии, робототехническое оборудование — вот что дает мощный эффект и хорошие результаты. Однако автоматизация сложных процессов нежелательна: неопределенность может привести к большому

количеству ошибок. Например, в сфере управления качеством важное значение имеет автоматизация анализа данных, собранных статистическими методами и.

2. **Рейнжиниринг «с чистого листа»** — существующий процесс полностью разрушается и утилизируется. Новый процесс создается с нуля путем фундаментального переосмысления существующего.

Очень трудно дать общий совет, как выполнить радикальную перестройку, которая заключается в том, чтобы все разрушить и начать с чистого листа. Результат существенно зависит от творческой активности, воображения, знаний, а также наличия современных технологий и людей, способных все это воплотить в жизнь. Каждый проект РБП уникален. Вот основные вопросы, которые требуют своего разрешения.

- Какие основные потребности мы хотим удовлетворить и для кого?

- Почему мы хотим удовлетворить эти потребности? Согласуется ли это с общей стратегией организации?

- Где надо удовлетворять эти потребности?

- Когда должны быть удовлетворены эти потребности?

- Каким образом будут удовлетворены эти потребности? Кто будет все это делать, какие нужны технологии?

Решающий фактор перестройки — творческая активность исполнителей. Чтобы разбудить воображение, дать импульс для творчества, задайте вашим сотрудникам следующие вопросы:

- Представьте себе, что вам нужно создать образ конкурента своей собственной организации. Каким бы вы его сделали, чтобы достичь лучшего результата?

- Как должен выглядеть идеальный процесс?

- Если бы вы могли создать всю вашу организацию заново с чистого листа, то как бы тогда выглядела эта организация и ее конкретный процесс?

Последнее, что нужно сделать на этапе выдвижения содержательных предложений вне зависимости от выбранного способа применения метода, — разработать перечень рекомендуемых изменений в результате перестройки бизнес-процесса.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 3. Разработка документации информационных систем

Практическое занятие № 5

Проектирование спецификации информационной системы индивидуальному заданию

Цель: получение навыков проектирования спецификации информационной системы

Выполнив работу, Вы будете:

уметь:

- У12. разрабатывать проектную документацию на эксплуатацию информационной системы;
- У13. использовать стандарты при оформлении программной документации.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства.

Задание:

1. Сформулировать: цели; задачи; требования к информационной системе.

Порядок выполнения работы:

1. Сформулировать цели и задачи создания информационной системы (см. практическая работа № 1). Охарактеризовать вид информационной системы, её назначение, используемые в работе системы данные. Сформулировать концептуальные требования к информационной системе.
2. Дать характеристику типового объекта автоматизации (организации, предприятия) для которого создаётся и на котором будет внедрена информационная система. Описать автоматизируемые бизнес-процессы.
3. Сформулировать требования к системе в целом. Описать структуру информационной системы. Перечислить функциональные подсистемы.
4. Сформулировать функциональные требования. Описать требования к функциям и задачам, выполняемым системой. Описать назначение и состав функций каждой из подсистем.
5. Описать предметную область. Разработать концептуальную модель данных предметной области. Сформулировать требования к информационному обеспечению системы.
6. Сформулировать требования к программному обеспечению системы. Описать требования к пользовательскому интерфейсу. Сформулировать технические требования к реализации и режимам работы информационной системы.
7. Используя полученные результаты, подготовить документ «Техническое задание на создание информационной системы», включающий в себя полное описание концептуальных, функциональных и технических требований к создаваемой системе.

Ход работы:**Стандарты, регламентирующие документирование**

Создание и применение АС сопровождается документированием обеспечения интерфейса с пользователями, а также возможности их освоения и функционального совершенствования. Комплексы отечественных стандартов, регламентирующие документирование АС на различных стадиях его создания, представляют во многом морально устаревшие стандарты серий «Информационная технология», «Единая система стандартов автоматизированной системы управления», «Единая система программной документации» [5]:

- ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
- ГОСТ 34.201-89 Информационная технология, комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначения документов при создании автоматизированных систем;
- РД 50-34.698-90 Информационная технология. Комплекс стандартов и руководящих указаний на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов;
- ГОСТ 24.101-80 Система технической документации на АСУ. Виды и комплектность документов;
- ГОСТ 24.102-80 Система технической документации на АСУ. Обозначение документов;
- ГОСТ 24.103-84 Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Основные положения;
- ГОСТ 24.202-80 Система технической документации на АСУ. Требования к содержанию документа «Технико-экономическое обоснование создания АСУ»;
- ГОСТ 24.203-80 Система технической документации на АСУ. Требования к содержанию общесистемных документов;

- ГОСТ 24.204-80 Система технической документации на АСУ. Требования к содержанию документа «Описание постановки задачи»;
- ГОСТ 24.205-80 Система технической документации на АСУ. Требования к содержанию документов по информационному обеспечению;
- ГОСТ 24.206-80 Система технической документации на АСУ. Требования к содержанию документов по техническому обеспечению;
- ГОСТ 24.207-80 Система технической документации на АСУ. Требования к содержанию документов по программному обеспечению;
- ГОСТ 24.208-80 Система технической документации на АСУ. Требования к содержанию документов стадии «Ввод в эксплуатацию»;
- ГОСТ 24.209-80 Система технической документации на АСУ. Требования к содержанию документов по организационному обеспечению;
- ГОСТ 24.210-82 Система технической документации на АСУ. Требования к содержанию документов по функциональной части;
- ГОСТ 24.211-82 Система технической документации на АСУ. Требования к содержанию документа «Описание алгоритма»;
- ГОСТ 24.301-80 Система технической документации на АСУ. Общие требования к выполнению текстовых документов;
- ГОСТ 24.302-80 Система технической документации на АСУ. Общие требования к выполнению схем;
- ГОСТ 24.303-80 Система технической документации на АСУ. Обозначения условные графические технических средств;
- ГОСТ 24.304-82 Система технической документации на АСУ. Требования к выполнению чертежей;
- ГОСТ 24.401-80 Система технической документации на АСУ. Внесение изменений;
- ГОСТ 24.401-80 Система технической документации на АСУ. Учет, хранение и обращение.

В свою очередь, международные стандарты регламентируют главным образом, документирование ПО. К ним относятся:

- ISO 6592:1986. ОИ. Руководство по документации для вычислительных систем;
- ISO 9294:1990-ТО. ИТ. Руководство по управлению документированием программного обеспечения;
- ISO 9127. ИТ. Пользовательская и рекламная документация на пакеты программ.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 3. Разработка документации информационных систем

Практическое занятие № 6

Разработка общего функционального описания программного средства по индивидуальному

заданию

Цель: получение навыков разработки общего функционального описания программного средства
Выполнив работу, Вы будете:

уметь:

- У12. разрабатывать проектную документацию на эксплуатацию информационной системы.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

Разработать общий функционал описания программного средства.

Порядок выполнения работы:

Подготовить исходные данные для разработки информационной системы (см. практическая работа № 1). Исходными данными для планирования являются:

общее описание некоторой информационной системы (назначение, область применения, решаемые задачи, технологические особенности реализации и внедрения);

Ограничения и условия разработки (требования заказчика, возможности команды разработчиков, сроки разработки, бюджет проекта и т.д.).

1. Составить эскизный план разработки информационной системы.

Составить документ «Технический проект» с описанием проектных решений (архитектура системы, логическая структура базы данных, решения по реализации пользовательского интерфейса и т.д.).

Составить документ «План тестирования» с описанием методики тестирования и контрольных тестов.

2. Составить документ «План ввода информационной системы в эксплуатацию».

Ход работы:

Стандарты, регламентирующие документирование

Создание и применение АС сопровождается документированием обеспечения интерфейса с пользователями, а также возможности их освоения и функционального совершенствования. Комплексы отечественных стандартов, регламентирующие документирование АС на различных стадиях его создания, представляют во многом морально устаревшие стандарты серий «Информационная технология», «Единая система стандартов автоматизированной системы управления», «Единая система программной документации» [5]:

- ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
- ГОСТ 34.201-89 Информационная технология, комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначения документов при создании автоматизированных систем;
- РД 50-34.698-90 Информационная технология. Комплекс стандартов и руководящих указаний на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов;
- ГОСТ 24.101-80 Система технической документации на АСУ. Виды и комплектность документов;
- ГОСТ 24.102-80 Система технической документации на АСУ. Обозначение документов;
- ГОСТ 24.103-84 Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Основные положения;

- ГОСТ 24.202-80 Система технической документации на АСУ. Требования к содержанию документа «Технико-экономическое обоснование создания АСУ»;
- ГОСТ 24.203-80 Система технической документации на АСУ. Требования к содержанию общесистемных документов;
- ГОСТ 24.204-80 Система технической документации на АСУ. Требования к содержанию документа «Описание постановки задачи»;
- ГОСТ 24.205-80 Система технической документации на АСУ. Требования к содержанию документов по информационному обеспечению;
- ГОСТ 24.206-80 Система технической документации на АСУ. Требования к содержанию документов по техническому обеспечению;
- ГОСТ 24.207-80 Система технической документации на АСУ. Требования к содержанию документов по программному обеспечению;
- ГОСТ 24.208-80 Система технической документации на АСУ. Требования к содержанию документов стадии «Ввод в эксплуатацию»;
- ГОСТ 24.209-80 Система технической документации на АСУ. Требования к содержанию документов по организационному обеспечению;
- ГОСТ 24.210-82 Система технической документации на АСУ. Требования к содержанию документов по функциональной части;
- ГОСТ 24.211-82 Система технической документации на АСУ. Требования к содержанию документа «Описание алгоритма»;
- ГОСТ 24.301-80 Система технической документации на АСУ. Общие требования к выполнению текстовых документов;
- ГОСТ 24.302-80 Система технической документации на АСУ. Общие требования к выполнению схем;
- ГОСТ 24.303-80 Система технической документации на АСУ. Обозначения условные графические технических средств;
- ГОСТ 24.304-82 Система технической документации на АСУ. Требования к выполнению чертежей;
- ГОСТ 24.401-80 Система технической документации на АСУ. Внесение изменений;
- ГОСТ 24.401-80 Система технической документации на АСУ. Учет, хранение и обращение.

В свою очередь, международные стандарты регламентируют главным образом, документирование ПО. К ним относятся:

- ISO 6592:1986. ОИ. Руководство по документации для вычислительных систем;
- ISO 9294:1990-ТО. ИТ. Руководство по управлению документированием программного обеспечения;
- ISO 9127. ИТ. Пользовательская и рекламная документация на пакеты программ.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 3. Разработка документации информационных систем

Практическое занятие № 7

Разработка руководства по установке программного средства по индивидуальному заданию

Цель: получение навыков разработки руководства по установке программного средства

Выполнив работу, Вы будете:

уметь:

- У12. разрабатывать проектную документацию на эксплуатацию информационной системы;
- У13. использовать стандарты при оформлении программной документации.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

Разработать руководство по установке программного средства для заданной информационной системы (см. практическая работа №1).

Порядок выполнения работы:

На примере руководства пользователя по установке программного обеспечения

«Интегрированной системы охранно-пожарной сигнализации ПРИТОК-А» (<https://docplayer.ru/29482251-Ustanovka-programmnogo-obespecheniya-rukovodstvo-operatora.html>) составить руководство пользователя по своей предметной области.

Ход работы:

Введение

Данное руководство описывает процесс установки ПО ИС ОПС ПРИТОК-А 3.7 на отдельном компьютере.

Примечание: Перед началом процесса установки программного обеспечения (далее ПО) необходимо убедиться в том, что компьютер, на котором производится установка, соответствует всем требованиям, описанным в документе «Рекомендуемые требования к компьютерам».

Установка ПО Приток-А 3.7

Для установки системы может быть использован один из следующих файлов:

- Pritok_3.7_Setup_Full.exe — установочный файл содержащий ПО, файлы документации на систему, примеры и т.д. - максимальный установочный пакет;

- Pritok_3.7_Setup.exe — установочный пакет без документации на систему;

- Pritok_3.7_Docs_Setup.exe — установочный пакет документации на систему.

Запустите нужный файл и следуйте инструкциям установки.

С помощью вышеперечисленных файлов можно выполнить как установку системы впервые так и обновление системы.

Описание процесса установки

1. Начало процесса установки. В данном окне установки вы получаете информацию о том, какая версия и номер сборки программного обеспечения ПРИТОК-А будет установлена на ваш компьютер. На примере версия 3.7.0 сборка 2273. Для продолжения установки нажмите кнопку "Далее". Для выхода из программы установки нажмите "Отмена" (см. Рис. 1).

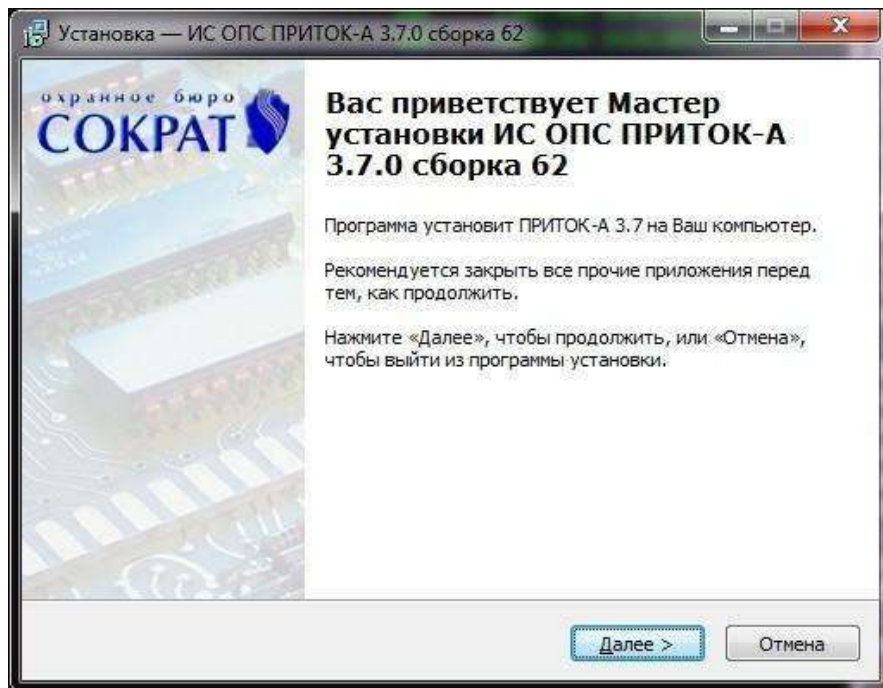


Рис 1: Приветствие, первое окно установки Лицензионное соглашение.

В окне установки «Лицензионное соглашение» (см. Рис. 2) представлен текст договора на предоставление права на использование программных продуктов ООО ОБ «Сократ». Пожалуйста, ознакомьтесь с договором и выберите пункт «Я принимаю условия лицензионного соглашения». Для продолжения установки нажмите кнопку "Далее". В ходе всего процесса установки для возврата на предыдущий шаг установки нажмите кнопку «Назад». Для выхода из программы установки нажмите кнопку «Отмена».

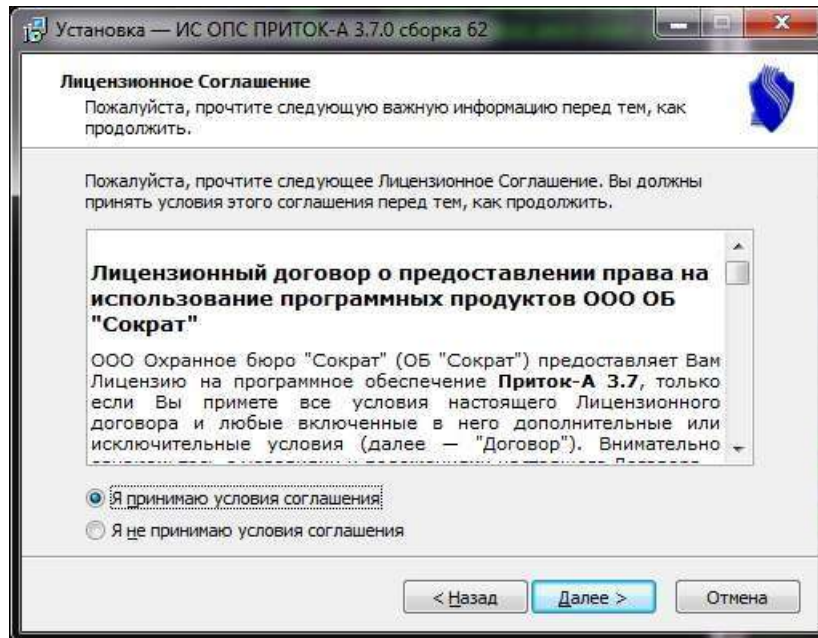


Рис 2: Лицензионное соглашение

2. *Выбор варианта установки.* На данном этапе внимательно прочитайте комментарии для разных вариантов установки и выберите нужный (см. Рис. 3). В том случае, если ПО Приток-А ни разу не устанавливалась на компьютере по умолчанию будет выбран вариант «Новая экспресс установка».

Ниже описанные шаги программы установки могут отличаться в зависимости от выбранного варианта установки. Рассмотрим ход инсталляции на примере варианта «Обычная установка\обновление».

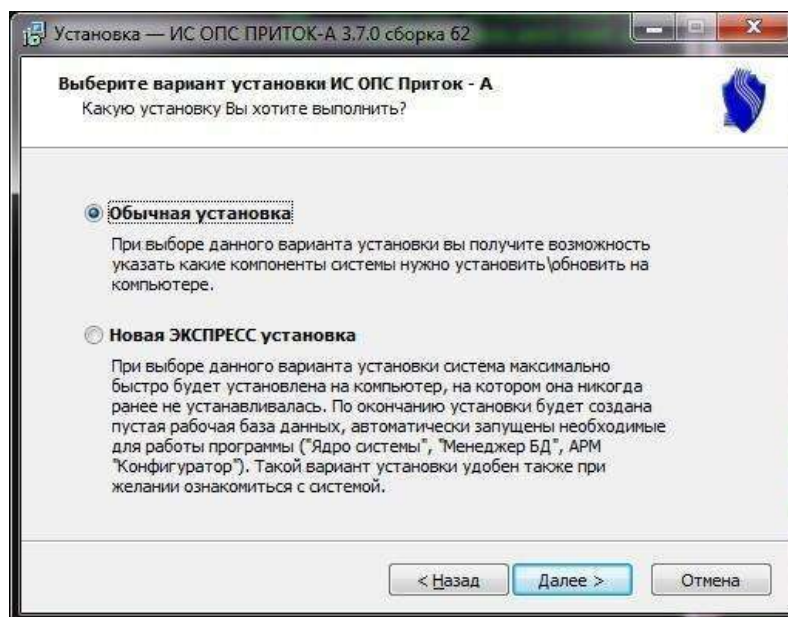


Рис 3: Выбор варианта установки

3. *Выбор директории, в которую будет произведена установка.* На данном этапе установки необходимо указать директорию, в которую будет установлено ПО. По умолчанию директорией для установки является "C:\Sokrat\Pritok-3.7\" (см. Рис. 4). При желании вы можете указать любой другой путь. В зависимости от версии Windows путь по умолчанию может отличаться. Для продолжения установки нажмите кнопку "Далее".

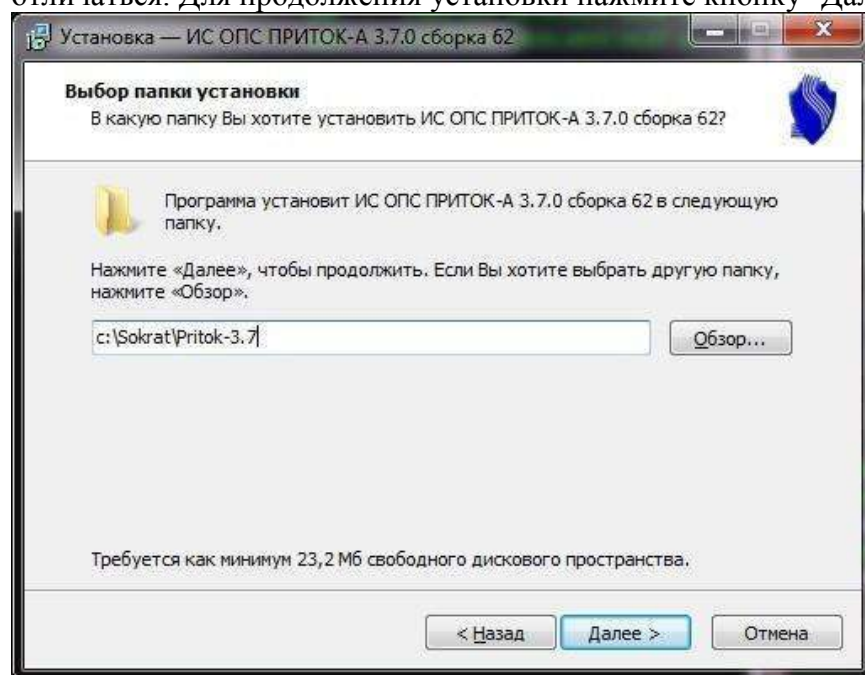


Рис 4: Выбор директории установки

4. *Копирование файлов для обновления.* Начиная с версии 3.6.0 сборки 2273 в состав Приток-А входит система обновления программного обеспечения. Более подробно о назначении и настройке системы обновления смотрите в инструкции «Система обновления программного обеспечения» (pritok_update.pdf).

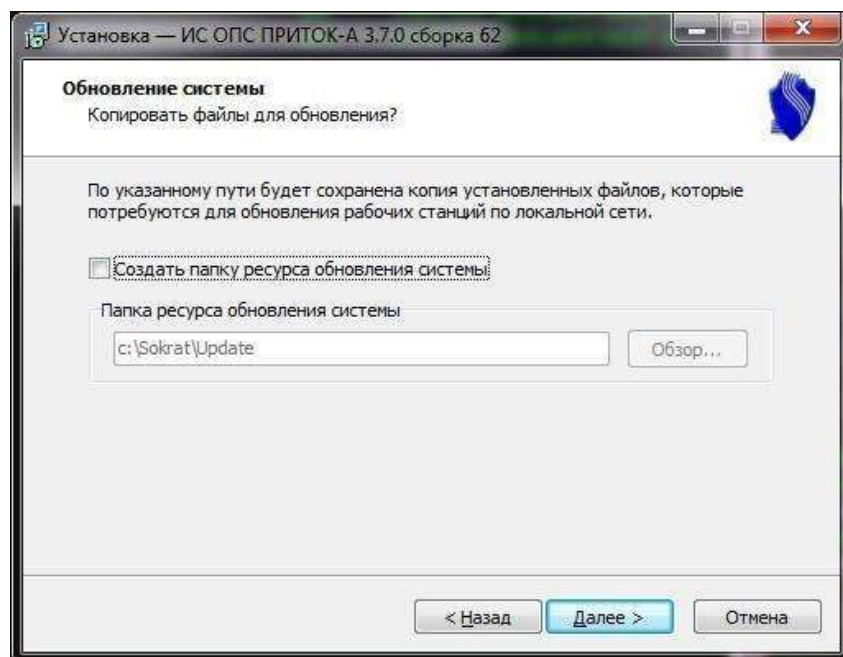


Рис 5: Обновление системы

В том случае, если необходимо создать папку для обновления программ для других компьютеров — ресурс обновления, установите галочку «Создать папку ресурса обновления системы» и укажите путь (см. Рис. 5). Программа установки скопирует файлы в основную директорию установки и в папку для обновления.

5. **Выбор типа установки.** На этом этапе выберите один из предложенных типов установки (см. Рис. 6):

- **Серверная установка** — установка серверной части системы для основного и резервного сервера (специально выделенный для работы серверной части системы компьютер).

- **Установка на рабочее место** — установка только клиентской части системы без серверных программ:

- **место администратора** — установка системы на рабочее место, на котором предполагается работа администратора базы данных (всей системы). Установка включает максимум клиентских программ системы;

- **место оператора ДПЦО** — установка системы на рабочее место, на котором предполагается работа дежурного офицера, оператора охраны, начальника караула и т.д.. В состав установки входит только программа АРМ ДПЦО;

- **место оператора МПО** — установка системы на рабочее место, на котором предполагается работа дежурного офицера МПО, оператора охраны МПО, начальника караула и т.д.. В состав установки входит только программа АРМ Приток-МПО (Мониторинг подвижных объектов) и необходимые для её работы компоненты.

- **Полная установка** — установка системы в максимальном варианте, который включает в себя установку и серверной и клиентской части системы. Такой тип установки может быть выбран когда необходимо установить всю систему на отдельно взятый компьютер.

Прочитайте комментарий для каждого типа установки и выберите нужный. Нажмите кнопку "Далее". Программа установки запомнит выбранный тип и предложит его при следующих запусках.

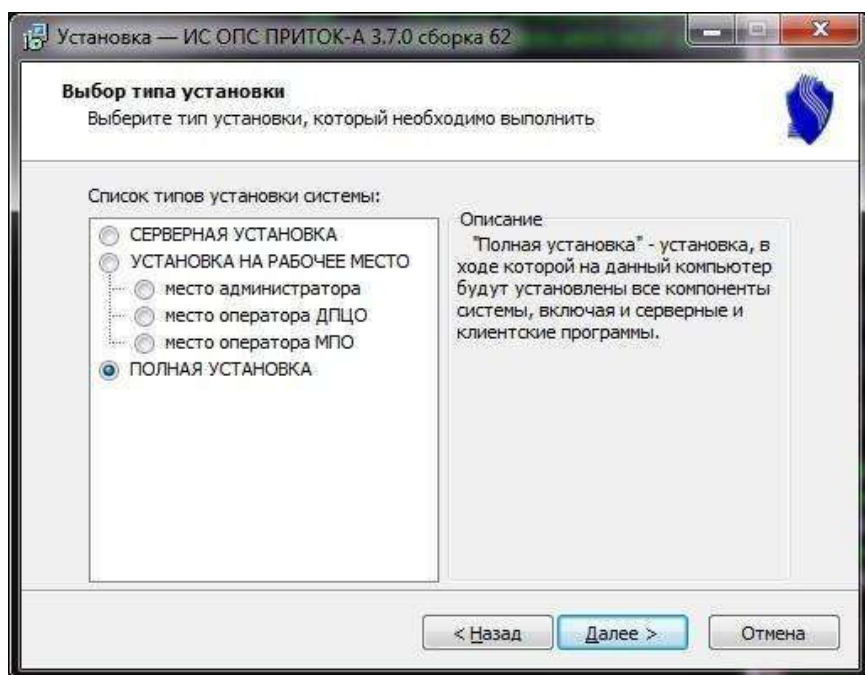


Рис 6: Выбор типа установки

6. *Выбор компонентов.* В окне выбора компонентов (см. Рис. 7) в списке перечислены все компоненты системы. Напротив некоторых из компонентов установлены галочки **согласно выбранному ранее типу установки**. Для того, чтобы изменить набор компонентов установите галочки напротив нужных. Для того, чтобы компонент не был установлен, уберите галочку напротив соответствующего компонента. Для установки нажмите кнопку "Далее". Не рекомендуется менять набор компонентов.

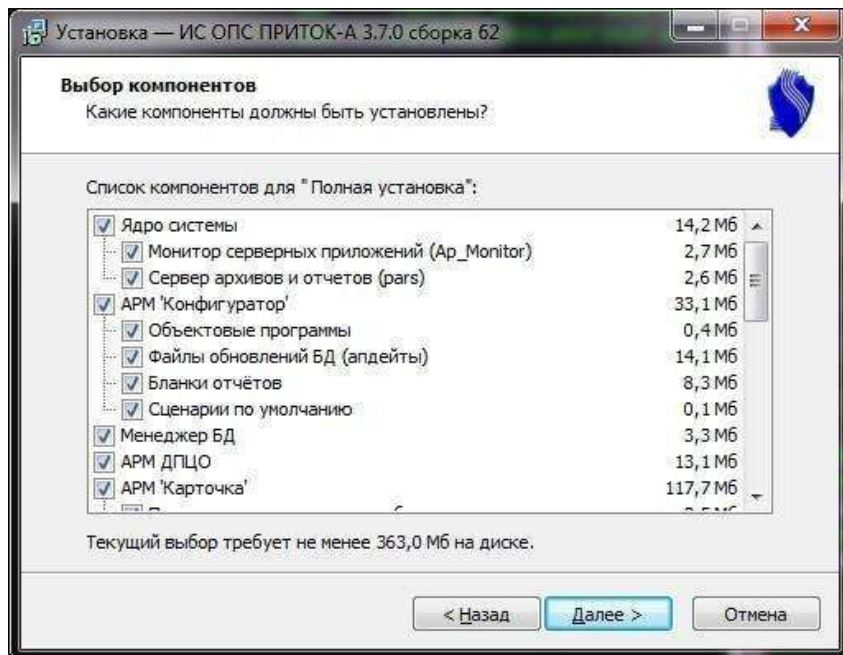


Рис 7: Выбор компонентов для типа установки

7. *Выбор папки в меню "Пуск".* Следующим шагом установки является указание программной группы, в которой будут созданы ярлыки для запуска программ. По умолчанию создаваемая группа называется "ПРИТОК-А 3.7" (см. Рис. 8). Вы можете изменить название создаваемой группы. В том случае, если вы не хотите создавать программную группу, то установите галочку напротив надписи "Не создавать папку в меню "Пуск"". Для продолжения установки нажмите кнопку "Далее".

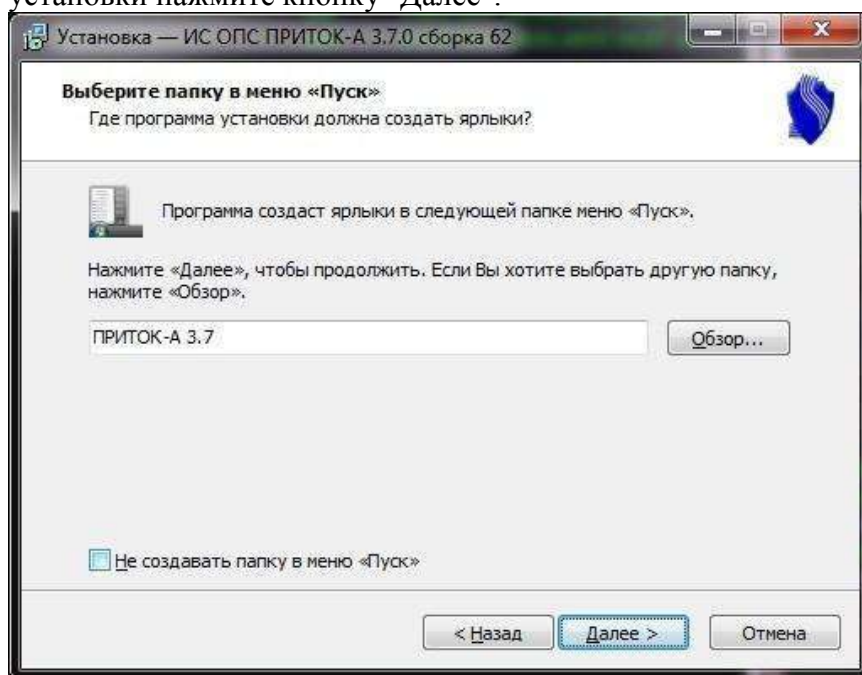


Рис 8: Выбор папки в меню "Пуск"

8. *Выбор дополнительных параметров установки.* На данном этапе установки необходимо определить необходимость создания ярлыков программ на рабочем столе. По умолчанию ярлыки программ будут созданы в папке с наименованием «Приток-А 3.7» (см. Рис. 9). Для продолжения установки нажмите кнопку "Далее".

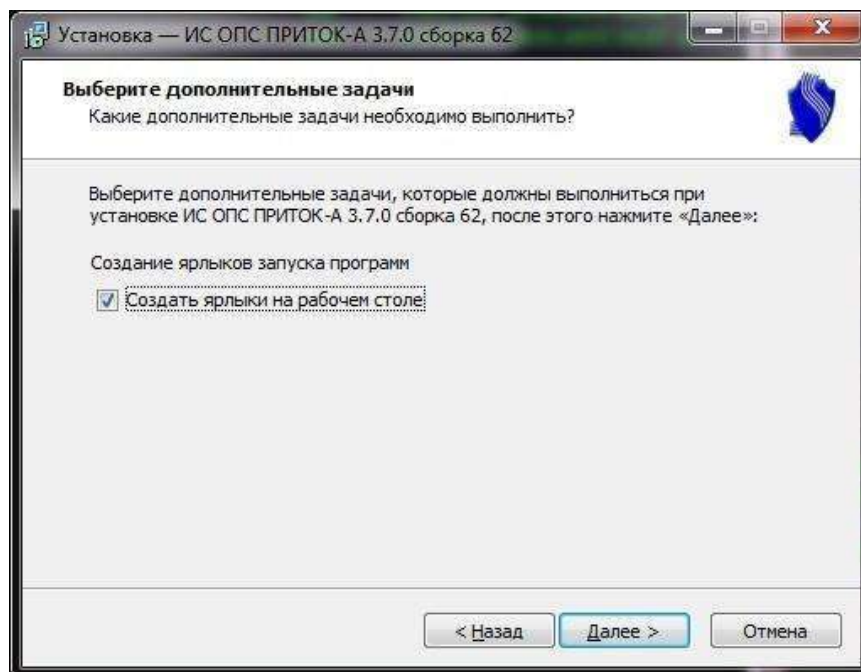


Рис 9: Создание ярлыков на рабочем столе

9. *Подготовка к установке.* На экран будет выдано окно с информацией о выбранном типе установки, выбранных компонентах и директории установки (см. Рис. 10). **Проверьте информацию и нажмите кнопку "Установить" для начала копирования файлов системы на компьютер.**

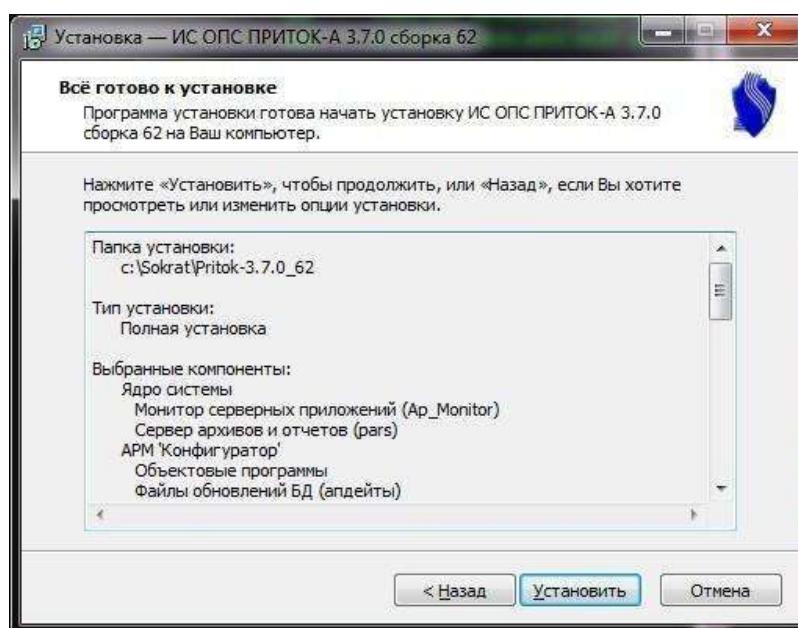


Рис 10: Все готово для установки системы

10. *Остановка запущенных программ.* На данном этапе программа установки принудительно закроет все работающие приложения системы Приток-А и остановит службы, если они есть. Список остановленных служб будет показан окне (см. Рис. 11).

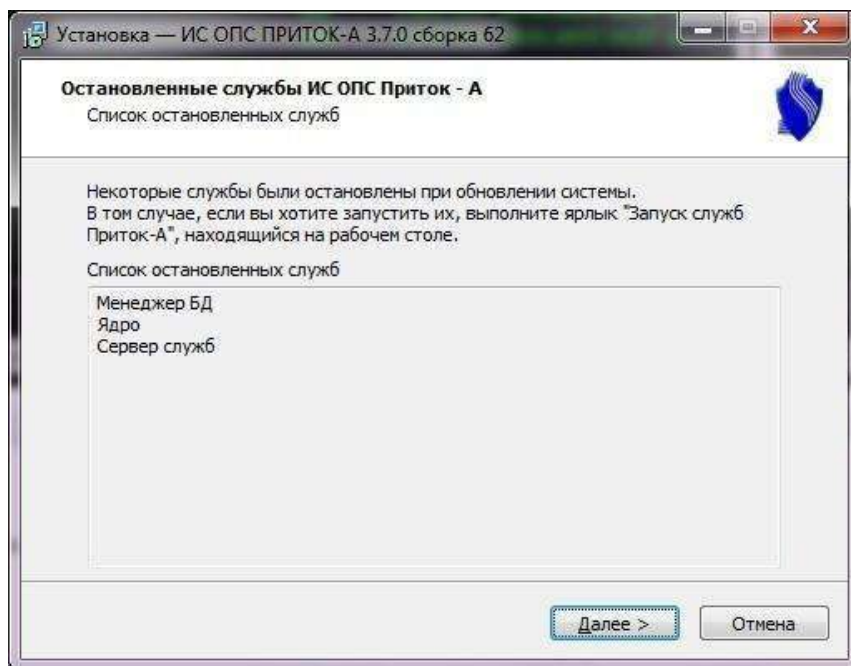


Рис 11: Список остановленных служб

Кроме этого, на рабочем столе компьютера будет создан ярлык для запуска остановленных служб с наименованием «Запуск служб Приток-А» (см. Рис. 12).

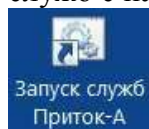


Рис 12: Ярлык

Для того, чтобы после окончания установки запустить остановленные службы выполните этот ярлык. После однократного выполнения ярлык будет автоматически удален.

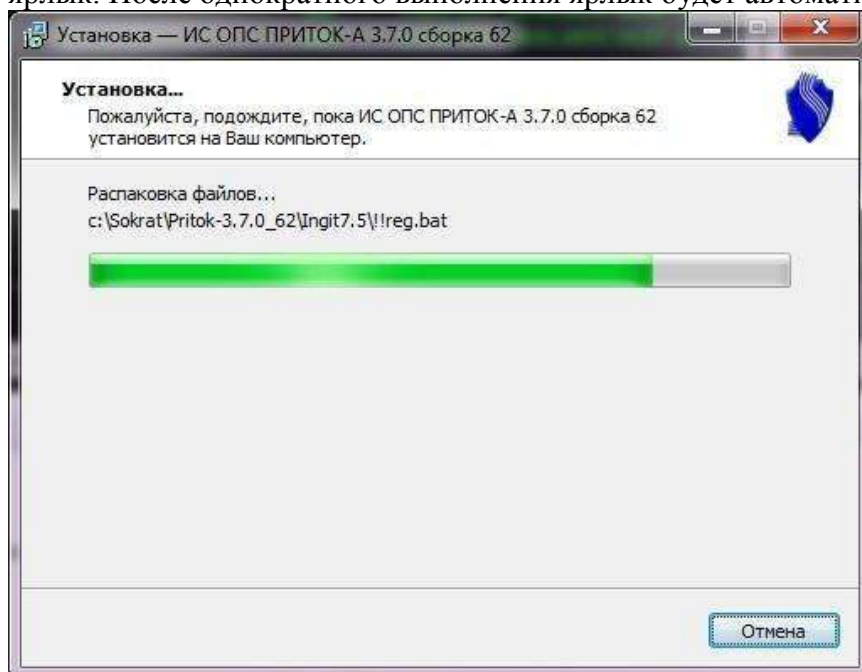


Рис 13: Копирование файлов на компьютер

11. *Копирование файлов.* На данном этапе инсталляции программа установки копирует необходимые файлы (см. Рис. 13).
12. *Завершающий этап установки.* На последнем этапе программа установки сообщит результат и предложит запустить программы **в зависимости от выбранного ранее типа**

установки (см. Рис. 14).



Рис 14: Завершение процесса установки

В том случае, если на компьютере не установлен необходимый для работы системы сервер баз данных «Firebird» или установленная версия отличается от нужной будет запущена инсталляция СУБД «Firebird» см. «Установка сервера баз данных Firebird».

Установка сервера баз данных Firebird

Сервер баз данных обязательно должен быть установлен на компьютере для правильной работы системы. Рассмотрим установку сервера баз данных Firebird на примере установки Firebird версии 2.0.5 под операционную систему Windows 7.

Внимание: Если на компьютере, на котором производится установка программного обеспечения, уже был ранее установлен Firebird какой-либо версии, то перед установкой версии 2.0.5 необходимо удалить старую версию и перезагрузить компьютер.

1. *Начало процесса установки сервера баз данных Firebird.* В данном окне установки вы получаете информацию о том, какая версия Firebird будет установлена на ваш компьютер. Для продолжения установки нажмите кнопку "Далее". Для выхода из программы установки нажмите "Отмена" (см. Рис. 15).

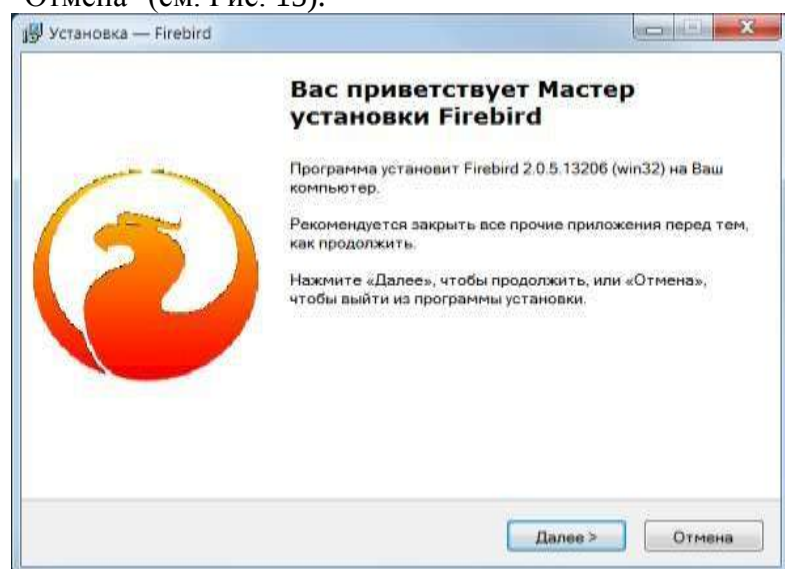


Рис 15: Начало процесса установки сервера БД Firebird

2. *Лицензионное соглашение БД Firebird.* На втором этапе установки сервера БД

приводится текст лицензионного соглашения с разработчиками. Для продолжения установки сделайте активным пункт "Я принимаю условия соглашения" и нажмите на кнопку "Далее". Для того, чтобы вернуться к предыдущему шагу установки нажмите "Назад". Для выхода из программы установки нажмите "Отмена" (см. Рис. 16).

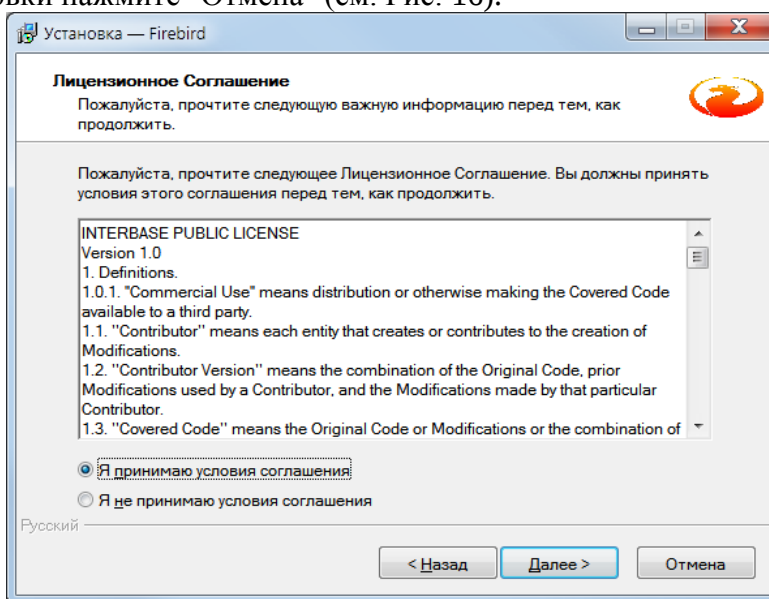


Рис 16: Лицензионное соглашение БД Firebird

3. Информация по установке. Для продолжения установки нажмите на кнопку "Далее".

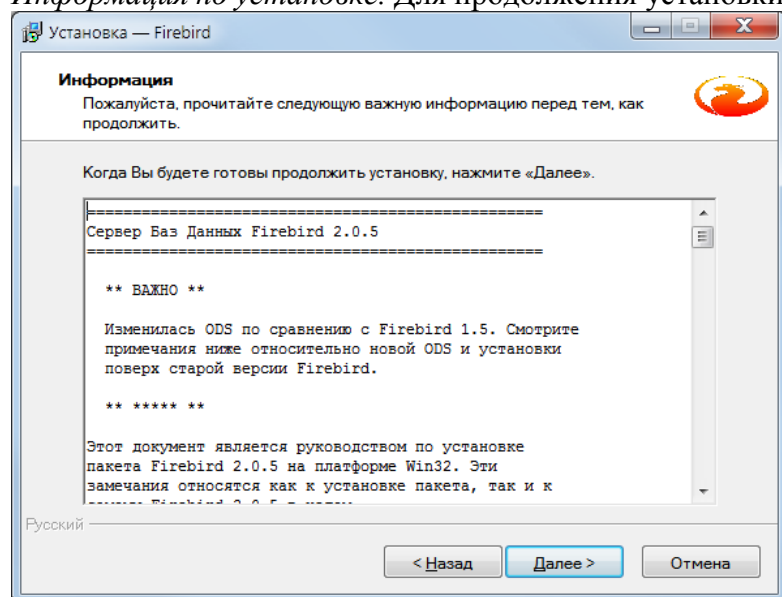


Рис 17: Информация по установке сервера БД Firebird

4. Выбор директории, в которую будет произведена установка. На данном этапе установки необходимо указать директорию, в которую будет установлен Firebird. По умолчанию директорией для установки является "C:\Program Files\Firebird\Firebird_2_0" (см. Рис. 18). При желании вы можете указать любой другой путь установки. Менять путь не рекомендуется. Для продолжения установки нажмите на кнопку "Далее"

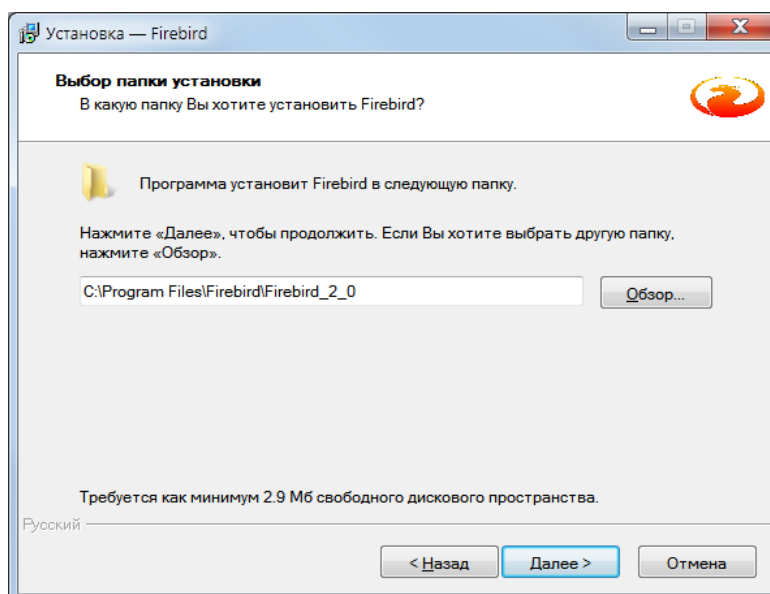


Рис 18: Выбор директории установки

5. *Определение типа устанавливаемого сервера БД Firebird.* На данном этапе установки необходимо указать какой тип сервера БД Firebird вы желаете установить. **Рекомендуется установка комплектации сервера «Классик».** Выберите пункт «Бинарные файлы ClassicServer» из списка (см. Рис. Ошибка: источник перекрёстной ссылки не найден). Для продолжения установки нажмите на кнопку "Далее".

6. *Выбор папки в меню "Пуск".* Следующим шагом установки является указание программной группы, в которой будут созданы значки для запуска программ. По умолчанию создаваемая группа имеет название "Firebird 2.0". Вы также можете изменить название создаваемой группы. В том случае, если вы не желаете создавать программную группу, то установите галочку напротив надписи "Не создавать папку в меню «Пуск»" (см. Рис. 19). Для продолжения установки нажмите на кнопку "Далее".

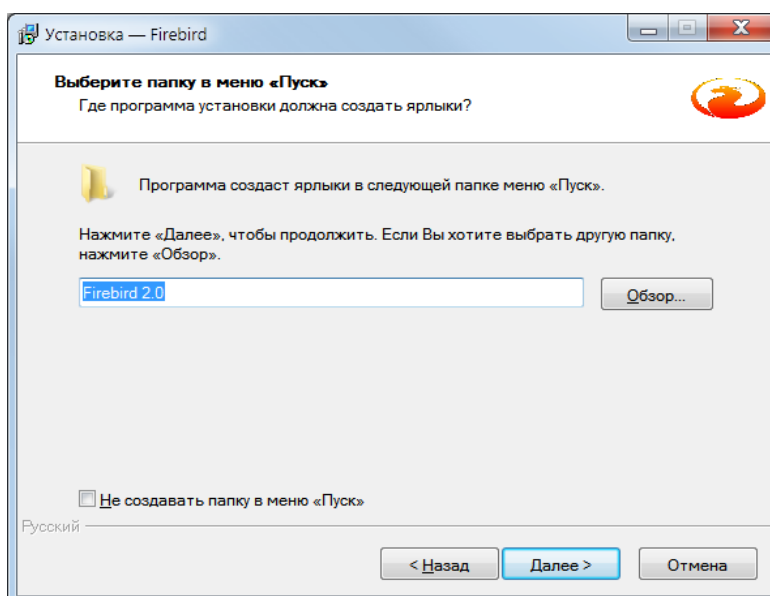


Рис 19: Выбор папки в меню "Пуск"

7. *Выбор дополнительных параметров установки.* На данном этапе установки необходимо определить дополнительные параметры установки и параметры последующих запусков сервера БД Firebird. Рекомендуемая конфигурация параметров изображена на рисунке (см. Рис. 20).

Обязательно выберите пункты:

- «Запускать в качестве Службы»

- «Автоматически запускать Firebird при каждой загрузке»
- «Копировать клиентскую библиотеку Firebird в каталог <system>»
- «Создать GDS32.DLL для поддержки унаследованных приложений». Для продолжения установки нажмите на кнопку "Далее".

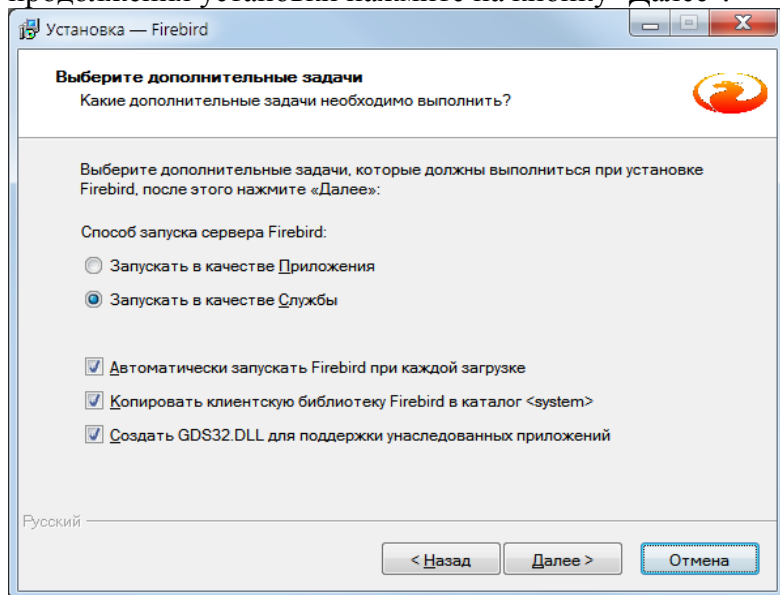


Рис 20: Выбор дополнительных параметров установки

8. *Окно проверки параметров установки.* Данный этап установки информирует о тех параметрах, которые были выбраны на предыдущих шагах установки. Для продолжения установки нажмите на кнопку "Установить". Для того, чтобы вернуться к предыдущему шагу установки нажмите "Назад". Для выхода из программы установки нажмите "Отмена" (см. Рис. 21).

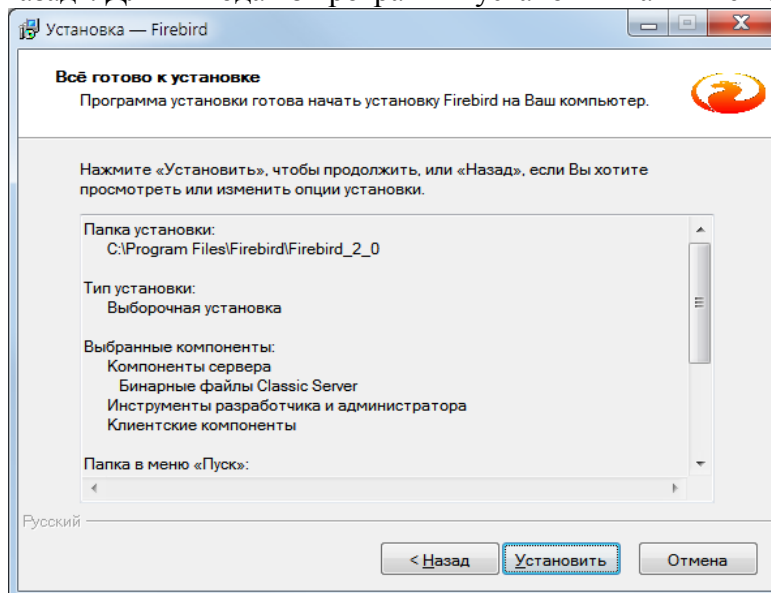


Рис 21: Проверка параметров установки

9. *Копирование файлов.* На данном этапе инсталляции программа установкископирует необходимые файлы (см. Рис. 22).

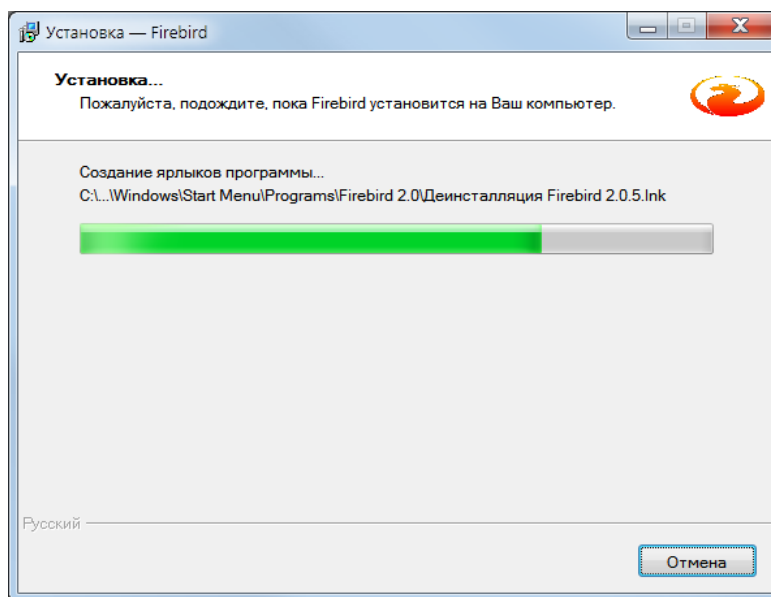


Рис 22: Копирование необходимых файлов

10. *Дополнительная информация об установленном продукте.* Прочитайте текст от разработчиков Firebird и нажмите кнопку "Далее" (см. Рис. 23).

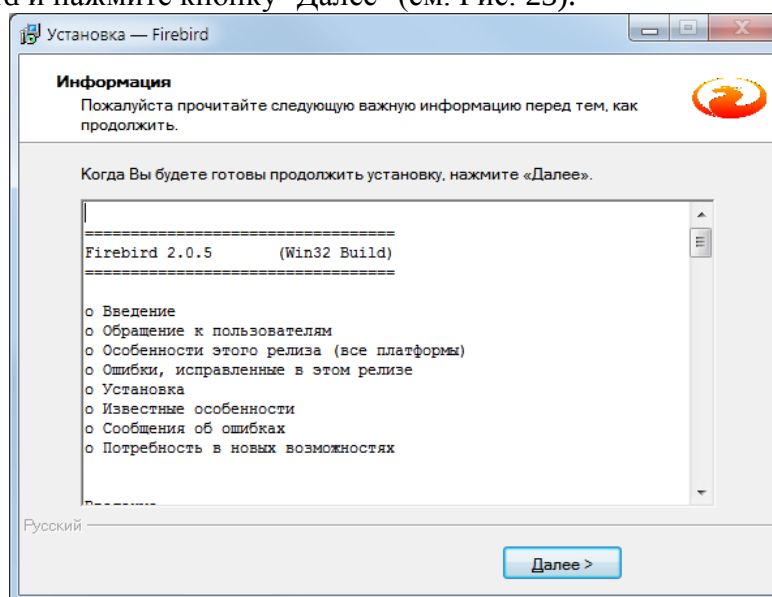


Рис 23: Информация о продукте

11. *Завершение процесса установки сервера БД Firebird.* На последнем этапе программа установки сообщит вам результат процесса установки и предложит запустить Firebird (рекомендуется). Для завершения процесса установки нажмите кнопку "Завершить" (см. Рис. 24).



Рис 24: Завершение процесса установки

Проверка работы сервера Firebird

После установки сервер Firebird должен быть запущен как служба (сервис) Windows. Для того, чтобы убедиться в этом откройте Панель управления\Администрирование\ Службы\ (ControlPanel\AdministrativeTools\Services). Служба Firebird должна выглядеть как показано на Рис. 25. Служба запускается автоматически и в данный момент времени работает. Внешний вид может изменяться в зависимости от версии Windows.

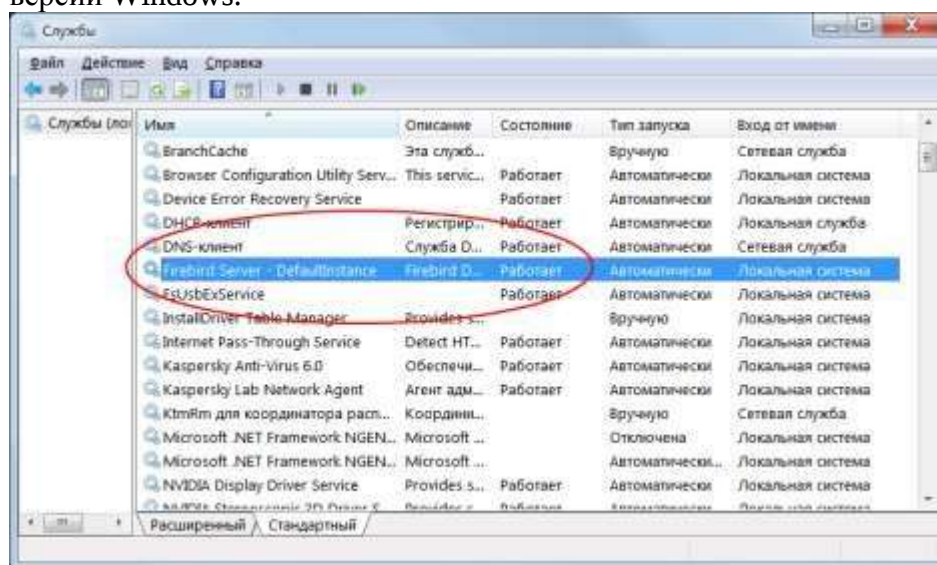


Рис 25: Проверка работы сервера

Настройка работы сервера Firebird

Рекомендуется обязательно произвести настройку сервера Firebird. Если пропустить этот шаг, то в дальнейшем по мере увеличения количества одновременно работающих клиентских программ системы возможны сбои в работе. Признаками возникновения такой ситуации являются ошибки при попытке запуска программ, появление в файле ошибок ядра записей вида

« Ошибка: -

>connectionrejectedbyremoteinterfaceConnectionnotestablishedErrorCode: 101». Для настройки следует выполнить следующие действия:

1. В списке служб выберите Firebird Server – DefaultInstance;
2. С помощью двойного щелчка мышью откройте свойства службы и перейдите на закладку «Вход в систему»;

3. Если галочка «Разрешить взаимодействие с рабочим столом» не установлена, то установите её.

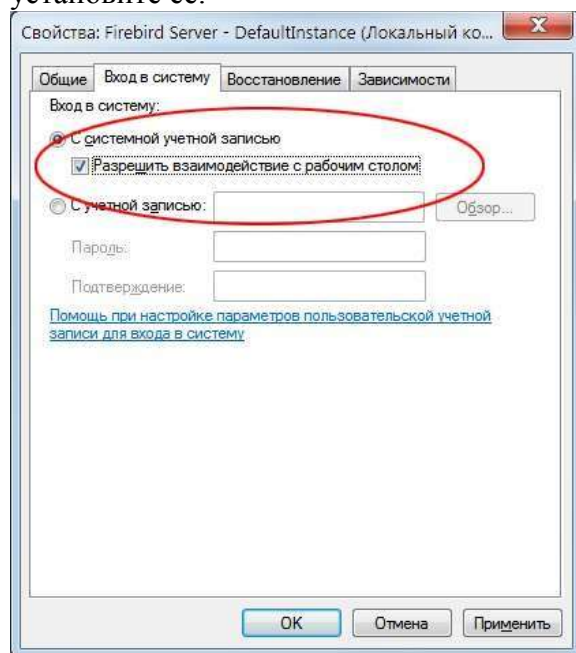


Рис 26: Свойства службы Firebird

Удаление ПО Приток-А 3.7

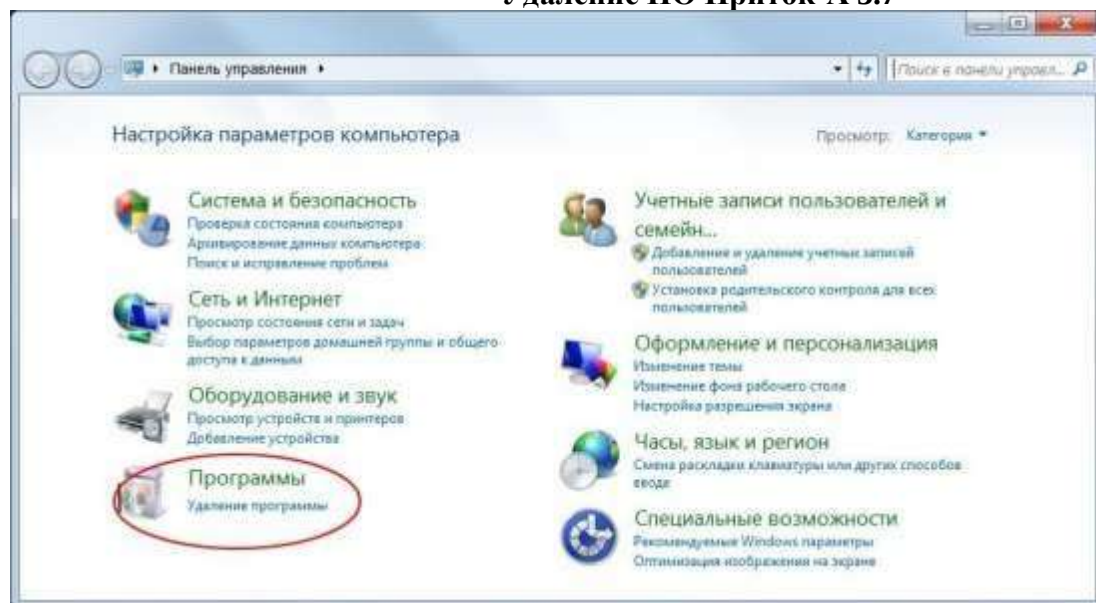


Рис 1: Панель управления

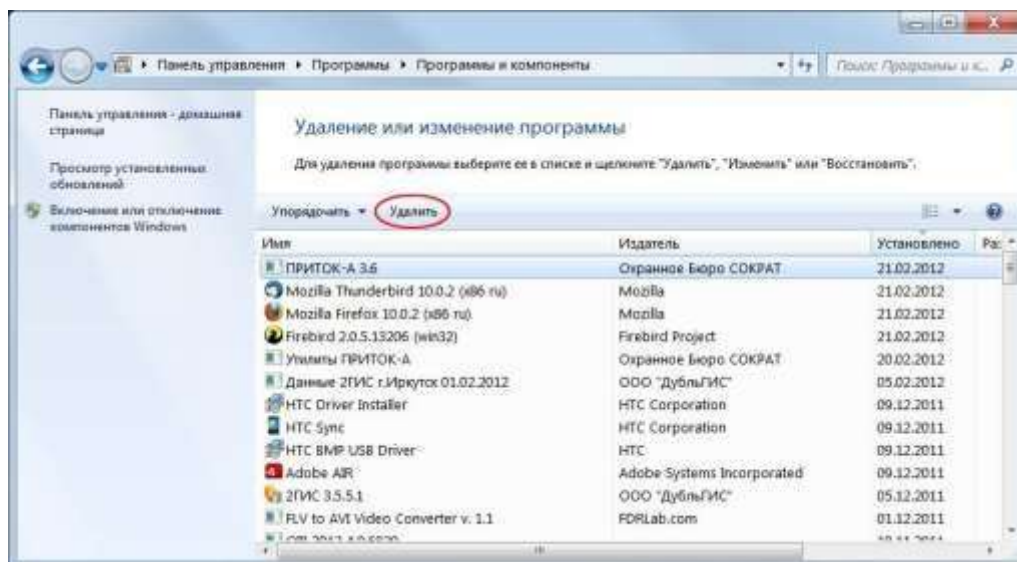


Рис 2: Удаление или изменение программ

Примечание:

- 1) Запущенные службы и программы ПО Приток-А будут остановлены автоматически и удалены.
- 2) Некоторые файлы могут остаться после выполнения удаления системы, удалите их в случае необходимости вручную.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 3. Разработка документации информационных систем

Практическое занятие № 8

Разработка руководства пользователя программного средства по индивидуальному заданию

Цель: получение навыков разработки руководства пользователя программного средства

Выполнив работу, Вы будете:

уметь:

- У12. разрабатывать проектную документацию на эксплуатацию информационной системы;
- У13. использовать стандарты при оформлении программной документации;
- У14. использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

1. Разработать руководство пользователя программного средства (см. практическая работа № 1)

Порядок выполнения работы:

На основе примера составить руководство пользователя.

Ход работы:

Стартовое окно программного продукта «Алгоритм оптимизации расчета себестоимости ремонта и модернизации асинхронного двигателя» изображено на рисунке 1.

EngineRM

Файл Справочники Результаты вычислений

Главное меню Счета Графики затрат

Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя

Тип двигателя: 5A200L4Y3

Выбор операции: Ремонт с модернизацией

Дата начала ремонта: 26.04.2016

Вид оптимизации: Без оптимизации

Дата окончания ремонта: 02.05.2016

Основные параметры двигателя

Специалисты

Основные операции

Ресурсы

Расходные коэффициенты

Расчет

Характеристика	Значение
Тип двигателя	5A200L4Y3
Мощность двигателя, кВт	45,00
Номинальная частота вращения, об/мин.	2940
Номинальная частота тока, Гц	50
Номинальный cos, о.е.	0,900
КПД, о.е.	93,400
Вес двигателя, кг.	310
Номинальное фазное напряжение, В	0,81
Номинальный ток, А	82
Схема соединения обмотки	Звезда

Добавить Изменить Удалить Информация

Результаты до оптимизации

Трудовые затраты, руб. 11183,75

Материальные затраты, руб. 25681,31

Общая стоимость, руб. 36865,06

Результаты после оптимизации (Минимизация себестоимости)

Трудовые затраты, руб. 10639,25

Материальные затраты, руб. 25681,31

Общая стоимость, руб. 36320,56

Рисунок 1 – Рабочее окно программы

После запуска программы необходимо выбрать тип двигателя из имеющихся в базе данных. Далее пользователю будут представлены все основные характеристики данного двигателя. Более детально характеристики можно посмотреть, нажав кнопку «Информация» или двойным нажатием мыши на таблицу характеристик. На рисунке 2 представлены все имеющиеся характеристики двигателя. Если ремонт предполагает занесения нового двигателя, ранее не ремонтировавшегося в компании «Энергосбережение - +», то пользователю следует выбрать кнопку «Добавить». После, открывается окно, обязательными полями для ввода является блок основных характеристик АД и поля блока параметров магнитной системы статора, отмеченные *, так как эти поля учувствуют в расчете себестоимости ремонта. Занесенный двигатель необходимо сохранить в базе данных, для этого нажимаем кнопку «Сохранить». Также предусмотрена возможность изменить имеющиеся характеристики двигателя, нажав кнопку «Изменить». Ненужный двигатель можно удалить из базы данных. Для этого нажимаем кнопку «Удалить». По умолчанию на главной странице программы открыта рабочая область с характеристиками двигателя.

Редактирование информации о двигателе

Основные параметры двигателя:		Параметры магнитной системы статора:	
Двигатель (*)	5A200L4Y3	Число обмоток (*)	3
Мощность двигателя, кВт (*)	45	Число пазов статора, Z1 (*)	48
Номинальная частота вращения, об/мин (*)	2940	Число пазов статора, Z2	38
Номинальная частота тока, Гц (*)	50	Диаметр статора, D1, мм (*)	392
Номинальная cos φ (*)	0,90	Диаметр статора D2, мм	287
Номинальный КПД, о.е. (*)	93,40	Число фаз	2
Вес двигателя, кг (*)	310	Число пар полюсов	8
Номинальное фазное напряжение, В (*)	0,81		
Номинальный ток, А (*)	82		

* обязательное поле

Сохранить Закрыть

Параметры обмотки статора:		Параметры обмотки статора:	
Схема соединения обмотки статора (*)	Звезда	Длина витка, lw, м	0,97
Тип обмотки 03 Шаг обмотки	1-8	Число параллельных ветвей, a	2
Число эф. проводников проводников в пазу	14+14	Диаметр провода рабочей обмотки, d1, мм	1,50
Число слоев рабочей обмотки	1	Диаметр провода рабочей обмотки, d'1, мм	1,58
Число слоев дополнительной обмотки	2	Диаметр провода дополнительной обмотки, d3, мм	2,00
Число элементарных проводников раб. обмотки	3	Диаметр провода дополнительной обмотки, d'3, мм	0,95
Число элементарных проводников доп. обмотки	1	Вес меди обмотки, кг	1,03
		Вес обмотки статора, кг	19,70

Рисунок 2 – Полный список параметров двигателя

После выбора двигателя, изменения его характеристик, если требуется, или добавления нового двигателя в базу данных, выбираем операцию. Пользователю предлагается либо производить расчет себестоимости ремонта, либо производить расчет себестоимости модернизации. Для того, чтобы произвести выбор типа операции, нужно жолкнуть левой кнопкой мыши по выпадающему списку «Выбор операции».

После задаем директивные сроки ремонта и модернизации асинхронного двигателя. Для этого в поле «Дата начала ремонта» задаем день, когда двигатель поступит на фирму.

Помимо расчета себестоимости ремонта и модернизации в программном продукте предусмотрен дополнительный модуль оптимизации затрат на выполнения операции, которую выбрал пользователь – это ремонт или модернизация. Предлагается три вариант:

- расчет себестоимости без оптимизации. Данный вариант предусматривает выполнения всех технологических операций последовательно с привлечением минимального количества специалистов. Все расчеты производятся по среднестатистическим данным;
- расчет себестоимости с оптимизаций по материальным затратам. Данный вариант оптимизации предусматривает понижение стоимости ремонта и модернизации за счет привлечения поставщика, у которого стоимость материальных ресурсов наименьшая. Фирма «Энергосбережение - +» сотрудничает с тремя поставщиками материальных ресурсов. При расчете затрат на проведения технологической операции пользователю предлагается поставщик, чья стоимость ресурса минимальна. За счет этого, происходит сокращения материальных затрат на ремонт и модернизация асинхронного двигателя;
- расчет себестоимости с оптимизаций «срочный ремонт». В данном варианте оптимизации предусматривается сокращение не денежных затрат, а сокращение временных затрат. Это достигается за счет распараллеливание технологических операций. Ряд операций способны происходить параллельно, т.е. они не зависят друг от друга. За счет этого время, затраченное на ремонт асинхронного двигателя, сокращается. Однако при данной оптимизации выбирается специалист не с минимальной стоимостью трудового часа, как в оптимизации «минимизация себестоимости», а свободный специалист. Поэтому, некоторые технологические операции может выполнять специалист с более высокой квалификацией. Из-за этого, стоимость ремонт или модернизации возрастает.

Поле «Дата окончания ремонта» заполниться автоматически, после расчёта себестоимости

ремонт или модернизации асинхронного двигателя.

Для начала расчета задаем рабочий штат. Для этого нажимаем кнопку «Специалисты».

Открывается рабочее область со специалистами, рисунок 3.

EngineRM

Файл Справочники Результаты вычислений

Главное меню Счета Графики затрат

Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя

Тип двигателя: SA200L4Y3

Выбор операции: Ремонт с модернизацией

Дата начала ремонта: 26.04.2016

Вид оптимизации: Без оптимизации

Дата окончания ремонта: 02.05.2016

Основные параметры двигателя

Специалисты

Основные операции

Ресурсы

Расходные коэффициенты

Расчет

Список специалистов:

- Инженер-технолог по электроремонту
- Материально-ответственное лицо
- Научный руководитель
- Обмотчик, 4 разряд
- Обмотчик, 5 разряд
- Обмотчик, 6 разряд
- Технический руководитель, 4 разряд
- Технический руководитель, 5 разряд
- Технический руководитель, 6 разряд
- Электрослесарь, 4 разряд
- Электрослесарь, 5 разряд
- Электрослесарь, 6 разряд

Список ответственных исполнителей:

- Инженер-технолог по электроремонту
- Материально-ответственное лицо
- Научный руководитель
- Обмотчик, 4 разряд
- Обмотчик, 5 разряд
- Обмотчик, 6 разряд
- Технический руководитель, 4 разряд
- Технический руководитель, 5 разряд
- Технический руководитель, 6 разряд
- Электрослесарь, 4 разряд
- Электрослесарь, 5 разряд
- Электрослесарь, 6 разряд

Добавить Удалить

Результаты до оптимизации		Результаты после оптимизации (Минимизация себестоимости)	
Трудовые затраты, руб.	11183,75	Трудовые затраты, руб.	10639,25
Материальные затраты, руб.	25681,31	Материальные затраты, руб.	25681,31
Общая стоимость, руб.	36865,06	Общая стоимость, руб.	36320,56

Рисунок 3 – Выбор специалистов

По умолчанию пользователю предложены специалисты: инженер-технолог по электроремонту, материально-ответственное лицо, научный руководитель, обмотчик 4–6 разрядов, технический руководитель и электрослесари 4–6 разрядов. Данные специалисты хранятся в базе данных. Пользователь самостоятельно может выбрать штат специалистов, которые будут участвовать в ремонте. Помимо этого, имеется возможность внести дополнительных рабочих, нажав на кнопку «Добавить». В появившемся рабочем окне назначаем должность специалиста, заработную плату, а также обязательным условием для работы программы, является привязки специалиста к определенным технологическим операциям, которые он выполняет. Рисунке 4 иллюстрирует пример создания нового специалиста. При привязки специалиста к технологической операции, на экран выводится поле, где необходимо указать количество времени, затрачиваемое на выполнение данной операции. Для сохранения нового специалиста необходимо нажать на кнопку «Сохранить».

Добавление нового специалиста

Информация по специалисту:

Квалификация:

Стоимость нормочаса специалиста, руб.

Выполняемые операции:

- ☐ Прием АД
- ☐ Разборка АД
- ☐ Мойка вспомогательных элементов АД
- ☐ Снятие подшипников ротора
- ☐ Установка новых подшипников ротора
- ☐ Дефектация статора
- ☐ Разборка электрической схемы обмотки статора
- ☐ Вырубка лобовых части обмотки статора
- ☐ Загрузка статора в печь
- ☐ Отжиг статора
- ☐ Выгрузка статора из печи
- ☐ Экструзия обмотки из пазов статора
- ☐ Чистка, правка магнитной системы статора
- ☐ Инструментальное исследование качества магнитной системы
- ☐ Электромагнитный пересчет параметров обмотки
- ☐ Составление технологической карты ремонта
- ☐ Изготовление блока компенсирующих конденсаторов
- ☐ Изготовление элементов пазовой изоляции
- ☐ Изготовление клиньев
- ☐ Изготовление катушек обмотки статора
- ☐ Изготовление элементов изоляции лобовых частей обмотки
- ☐ Укладка обмотки статора, сбор эл.схемы
- ☐ Инструментальное испытание обмотки статора
- ☐ Загрузка статора в пропиточный бак
- ☐ Пропитка статора

Сохранить

Закрыть

Рисунок 4 – Добавление нового специалиста

Для отображения технологических операций, участвующих в ремонте и модернизации, необходимо нажать на кнопку «Основные операции». Так как в данной работе рассматривается капитальный ремонт асинхронного двигателя, все операции отмечены галкой. Для корректной работы программы предусмотрен обработчик, который следит за привязкой технологической операции к специалисту. Поэтому при возникновении ситуации, когда пользователь не назначил специалиста на конкретную технологическую операцию, т.е. операция «не прикрыта» никаким специалистом, на экране, в разделе «Информация», появляется сообщение «Отсутствует исполнитель по данной операции». Пользователю необходимо вернуться в раздел «Специалисты» и назначить необходимо рабочего. Определенные специалисты могут выполнять определенные действия. В базе данных хранится таблица, где связаны специалисты с операцией. Для удобства пользователю для каждой технологической операции предлагается список исполнителей. На рисунке 27 изображено рабочее окно «Основные операции». При выборе ремонта с оптимизацией «срочный ремонт» для пользователя на экране изображается граф, где видно, какие операции могут проходить параллельно, рисунок 5.

EngineRM

Файл Справочники Результаты вычислений

Главное меню Смета Графики затрат

Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя

Тип двигателя: 5A200L4Y3

Выбор операции: Ремонт с модернизацией

Дата начала ремонта: 26.04.2016

Вид оптимизации: Без оптимизации

Дата окончания ремонта: 02.05.2016

Основные параметры двигателя

Специалисты

Основные операции

Ресурсы

Расходные коэффициенты

Расчет

Основные операции при ремонте с модернизацией

- ☒ Прием АД
- ☒ Разборка АД
- ☒ Мойка вспомогательных элементов АД
- ☒ Снятие подшипников ротора
- ☒ Установка новых подшипников ротора
- ☒ Дефектация статора
- ☒ Разборка электрической схемы обмотки статора
- ☒ Вырубка лобовых части обмотки статора
- ☒ Загрузка статора в печь
- ☒ Отжиг статора
- ☒ Выгрузка статора из печи
- ☒ Экструзия обмотки из пазов статора
- ☒ Чистка, правка магнитной системы статора
- ☒ Инструментальное исследование качества
- ☒ Электромагнитный пересчет параметров обмотки
- ☒ Составление технологической карты ремонта
- ☒ Изготовление блока компенсирующих конденсаторов
- ☒ Изготовление элементов пазовой изоляции
- ☒ Изготовление клиньев
- ☒ Изготовление катушек обмотки статора
- ☒ Изготовление элементов изоляции лобовых частей
- ☒ Укладка обмотки статора, сбор эл. схемы
- ☒ Инструментальное испытание обмотки статора
- ☒ Загрузка статора в пропиточный бак

Назначенные специалисты:

Материально-ответственное лицо

Специалисты:

Материально-ответственное лицо

Добавить

Результаты до оптимизации		Результаты после оптимизации (Минимизация себестоимости)	
Трудовые затраты, руб.	11183,75	Трудовые затраты, руб.	10639,25
Материальные затраты, руб.	25681,31	Материальные затраты, руб.	25681,31
Общая стоимость, руб.	36865,06	Общая стоимость, руб.	36320,56

Рисунок 5 – Основные операции

EngineRM

Файл Справочники Результаты вычислений

Главное меню Смета Графики затрат

Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя

Тип двигателя: 5A200L4Y3

Выбор операции: Ремонт с модернизацией

Дата начала ремонта: 26.04.2016

Вид оптимизации: Без оптимизации

Дата окончания ремонта: 02.05.2016

Основные параметры двигателя

Специалисты

Основные операции

Ресурсы

Расходные коэффициенты

Расчет

Основные операции при ремонте с модернизацией

- ☒ Прием АД
- ☒ Разборка АД
- ☒ Мойка вспомогательных элементов АД
- ☒ Снятие подшипников ротора
- ☒ Установка новых подшипников ротора
- ☒ Дефектация статора
- ☒ Разборка электрической схемы обмотки статора
- ☒ Вырубка лобовых части обмотки статора
- ☒ Загрузка статора в печь
- ☒ Отжиг статора
- ☒ Выгрузка статора из печи
- ☒ Экструзия обмотки из пазов статора
- ☒ Чистка, правка магнитной системы статора
- ☒ Инструментальное исследование качества
- ☒ Электромагнитный пересчет параметров обмотки
- ☒ Составление технологической карты ремонта
- ☒ Изготовление блока компенсирующих конденсаторов
- ☒ Изготовление элементов пазовой изоляции
- ☒ Изготовление клиньев
- ☒ Изготовление катушек обмотки статора
- ☒ Изготовление элементов изоляции лобовых частей
- ☒ Укладка обмотки статора, сбор эл. схемы
- ☒ Инструментальное испытание обмотки статора
- ☒ Загрузка статора в пропиточный бак

Назначенные специалисты:

Материально-ответственное лицо

Специалисты:

Материально-ответственное лицо

Добавить

Результаты до оптимизации		Результаты после оптимизации (Минимизация себестоимости)	
Трудовые затраты, руб.	11183,75	Трудовые затраты, руб.	10639,25
Материальные затраты, руб.	25681,31	Материальные затраты, руб.	25681,31
Общая стоимость, руб.	36865,06	Общая стоимость, руб.	36320,56

Рисунок 6 – Сетевая модель ремонта

На рисунке 6 представлена сетевая модель модернизации асинхронного двигателя.

The screenshot shows the 'EngineRM' application window. The title bar includes 'EngineRM' and standard window controls. The menu bar has 'Файл', 'Справочники', and 'Результаты вычислений'. Below the menu is a sub-menu with 'Главное меню', 'Счета', and 'Графики затрат'. The main title is 'Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя'.

On the left, there are input fields for:

- Тип двигателя: 5A200L4Y3
- Выбор операции: Ремонт с модернизацией
- Дата начала ремонта: 26.04.2016
- Вид оптимизации: Без оптимизации
- Дата окончания ремонта: 02.05.2016

 Below these are buttons for 'Основные параметры двигателя', 'Специалисты', 'Основные операции' (highlighted in blue), 'Ресурсы', 'Расходные коэффициенты', and 'Расчет'.

The central 'Основные операции при ремонте с модернизацией' list includes:

- ☒ Прием АД
- ☒ Разборка АД
- ☒ Мойка вспомогательных элементов АД
- ☒ Снятие подшипников ротора
- ☒ Установка новых подшипников ротора
- ☒ Дефектация статора
- ☒ Разборка электрической схемы обмотки статора
- ☒ Вырубка лобовых части обмотки статора
- ☒ Загрузка статора в печь
- ☒ Отжиг статора
- ☒ Выгрузка статора из печи
- ☒ Экструзия обмотки из пазов статора
- ☒ Чистка, правка магнитной системы статора
- ☒ Инструментальное исследование качества
- ☒ Электромагнитный пересчет параметров обмотки
- ☒ Составление технологической карты ремонта
- ☒ Изготовление блока компенсирующих конденсаторов
- ☒ Изготовление элементов пазовой изоляции
- ☒ Изготовление клиньев
- ☒ Изготовление катушек обмотки статора
- ☒ Изготовление элементов изоляции лобовых частей
- ☒ Укладка обмотки статора, сбор эл. схемы
- ☒ Инструментальное испытание обмотки статора
- ☒ Загрузка статора в пропиточный бак

On the right, there are sections for 'Назначенные специалисты:' and 'Специалисты:', both with a 'Материально-ответственное лицо' field and a 'Добавить' button.

At the bottom, the 'Результаты до оптимизации' and 'Результаты после оптимизации (Минимизация себестоимости)' are shown in a table:

Результаты до оптимизации		Результаты после оптимизации (Минимизация себестоимости)	
Трудовые затраты, руб.	11183,75	Трудовые затраты, руб.	10639,25
Материальные затраты, руб.	25681,31	Материальные затраты, руб.	25681,31
Общая стоимость, руб.	36865,06	Общая стоимость, руб.	36320,56

Рисунок 7 – Основные операции

Вышеупомянутые вкладки соответствуют расчету трудовых затрат: специалисты, норма часа специалиста, привязка специалиста к операции и т.п. Для отображения материальных ресурсов, необходимых для проведения ремонта, а также указание стоимости ресурса, необходимо нажать кнопку «Ресурсы», рисунок 8. Помимо функции просмотра, пользователь имеет возможность добавить новый ресурс в базу данных, с указанием наименования, единицы измерения, а также стоимостью, а также внести корректировки в уже имеющийся ресурс, нажав на кнопку «Изменить», рисунок 8.

Помимо учета материальных затрат, необходимо знать расходный коэффициент по каждому материальному ресурсу. Расходный коэффициент определяется экспериментальным путем, поэтому для удобства и простоты работы пользователя все расходные коэффициенты имеют значение по умолчанию. Для просмотра таблицы с их значением, а также операции, в котором они участвует, и наименование материального ресурса, с которым используется коэффициент, рисунок 9. Пользователь имеет возможность редактировать расходный коэффициент, менять его значение. Для этого необходимо нажать на кнопку «Изменить», рисунок 9.

EngineRM

Файл Справочники Результаты вычислений

Главное меню Счета Графики затрат

Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя

Тип двигателя: 5A200L4Y3

Выбор операции: Ремонт с модернизацией

Дата начала ремонта: 26.04.2016

Вид оптимизации: Без оптимизации

Дата окончания ремонта: 02.05.2016

Основные параметры двигателя

Специалисты

Основные операции

Ресурсы

Расходные коэффициенты

Расчет

Ресурсы:

Наименование ресурса	Ед.изм.	Цена, руб
1 кВАр компенсатора реактивной мощности	кВАр	175
Гайка	шт.	1
Конструкционная листовая сталь	кг	45
Масляная краска	кг	130
Медный наконечник	шт.	12
Один эквивалентный клин	шт.	20
Пазовые электроизоляционные материалы	м^2	350
Подшипник	шт.	1500
Припой	кг	480
Провод выводных концов	м	12
Пропиточный лак	кг	115
Стеклолента для бандажирования	м	12

Добавить Изменить Удалить

Результаты до оптимизации

Трудовые затраты, руб. 11183,75

Материальные затраты, руб. 25681,31

Общая стоимость, руб. 36865,06

Результаты после оптимизации (Минимизация себестоимости)

Трудовые затраты, руб. 10639,25

Материальные затраты, руб. 25681,31

Общая стоимость, руб. 36320,56

Рисунок 8 – Материальные ресурсы

Редактирование информации о ресурсе

Наименование ресурса: 1 кВАр компенсатора реактивной мощности

Единица измерения: кВАр

Цена, руб.: 175

Сохранить Закрыть

Рисунок 9 – Редактирование материального ресурса

EngineRM

Файл Справочники Результаты вычислений

Главное меню Счета Графики затрат

Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя

Тип двигателя: 5A200L4Y3

Выбор операции: Ремонт с модернизацией

Дата начала ремонта: 26.04.2016

Вид оптимизации: Без оптимизации

Дата окончания ремонта: 02.05.2016

Основные параметры двигателя

Специалисты

Основные операции

Ресурсы

Расходные коэффициенты

Расчет

Таблица расходных коэффициентов ресурсов на соответствующие операции:

Операция	Расх.коэф.	Кол-во	Ресурс
Установка новых подшипников	1	2	Подшипник
Отжиг статора	1	20	Электроэнергия
Изготовление блока компенсир	1		1 кВАр компенсатора реактивн
Изготовление блока компенсир	1		Устройство защиты электрооб
Изготовление блока компенсир	1,1		Конструкционная листовая ста
Изготовление блока компенсир	1,1		Стеклотекстолит
Изготовление блока компенсир	1	24	Гайка
Изготовление блока компенсир	1	24	Шайба
Изготовление блока компенсир	1		Припой
Изготовление блока компенсир	1	6	Шпилька
Изготовление блока компенсир	1		Медный наконечник
Изготовление элементов пазов	1,2		Пазовые электроизоляционные

Добавить Изменить Удалить

Результаты до оптимизации

Трудовые затраты, руб. 11183,75

Материальные затраты, руб. 25681,31

Общая стоимость, руб. 36865,06

Результаты после оптимизации (Минимизация себестоимости)

Трудовые затраты, руб. 10639,25

Материальные затраты, руб. 25681,31

Общая стоимость, руб. 36320,56

Рисунок 10 – Расходный коэффициент

Редактирование информации о расходном коэффициенте

Операция: Установка новых подшипников ротора

Ресурс: Подшипник

Расходный коэффициент: 1

Количество ресурса, шт.: 2

Сохранить Закрыть

Рисунок 11 – Редактирование расходного коэффициента

Для расчета себестоимости ремонта или модернизации асинхронного двигателя необходимо нажать на кнопку «Расчет». На главной странице в вычисляемом поле появится стоимость ремонт или модернизации АД. Данная сумма складывается из двух составляющих – это трудовые и материальные затраты. В программном продукте предусмотрено разделение этих затрат, благодаря чему пользователь имеет наглядное представление, сколько потребуется средств для оплаты заработной платы специалистам, выполняющих технологические операции, а также сколько потребуется для оплаты материальных ресурсов. Расчет себестоимости ремонт или модернизации АД представлен на рисунке 12.

EngineRM

Файл Справочники Результаты вычислений

Главное меню Смета Графики затрат

Расчет себестоимости и оптимизация затрат на ремонт, модернизацию асинхронного двигателя

Тип двигателя: 5A200L4Y3

Выбор операции: Ремонт с модернизацией

Дата начала ремонта: 26.04.2016

Вид оптимизации: Без оптимизации

Дата окончания ремонта: 02.05.2016

Основные параметры двигателя

Специалисты

Основные операции

Ресурсы

Расходные коэффициенты

Расчет

Основные операции при ремонте с модернизацией

- ☒ Прием АД
- ☒ Разборка АД
- ☒ Мойка вспомогательных элементов АД
- ☒ Снятие подшипников ротора
- ☒ Установка новых подшипников ротора
- ☒ Дефектация статора
- ☒ Разборка электрической схемы обмотки статора
- ☒ Вырубка лобовых части обмотки статора
- ☒ Загрузка статора в печь
- ☒ Отжиг статора
- ☒ Выгрузка статора из печи
- ☒ Экструзия обмотки из пазов статора
- ☒ Чистка, правка магнитной системы статора
- ☒ Инструментальное исследование качества
- ☒ Электромагнитный пересчет параметров обмотки
- ☒ Составление технологической карты ремонта
- ☒ Изготовление блока компенсирующих конденсаторов
- ☒ Изготовление элементов пазовой изоляции
- ☒ Изготовление клиньев
- ☒ Изготовление катушек обмотки статора
- ☒ Изготовление элементов изоляции лобовых частей
- ☒ Укладка обмотки статора, сбор эл. схемы
- ☒ Инструментальное испытание обмотки статора
- ☒ Загрузка статора в пропиточный бак

Назначенные специалисты:

Материально-ответственное лицо

Специалисты:

Материально-ответственное лицо

Добавить

Результаты до оптимизации		Результаты после оптимизации (Минимизация себестоимости)	
Трудовые затраты, руб.	11183,75	Трудовые затраты, руб.	10639,25
Материальные затраты, руб.	25681,31	Материальные затраты, руб.	25681,31
Общая стоимость, руб.	36865,06	Общая стоимость, руб.	36320,56

Рисунок 12 – Результаты расчета себестоимости операции

Одним из функционалов работы программы является составление сметы трудовых и материальных затрат на проведение ремонта, вкладка «Смета». Под сметой понимается план расходов на осуществление ремонта или модернизации асинхронного двигателя с уточнением всех статей расходов для каждой технологической операции. При выводе сметы учитываются трудовые затраты, в которых пользователю предоставляется сводная таблица, где указано наименование технологической операции, время проведения каждой технологической операции, прикрепленный специалист с уточнением его заработной платы за выполнения данной операции, и материальные затраты, в которых учитывается материальный ресурс, его количество, стоимость и расходный коэффициент для данного ресурса. На рисунке 13 изображена смета на проведения ремонта асинхронного двигателя типа 5A200L4Y3. Помимо этого, в окне предусмотрен выбор сметы: смета без оптимизации ремонта и модернизации АД или смета с оптимизацией по заданному критерию, рисунок 13, 36.

EngineRM								
Файл Справочники Результаты вычислений								
Главное меню Смета Графики затрат								
Смета затрат на ремонт с модернизацией асинхронного двигателя:								
☒ Без оптимизации ☐ С оптимизацией (Минимизация себестоимости)								
Время ремонта: 50,6 часов. Стоимость ремонта: 36865,06 руб. (затраты трудовые - 11183,75 руб., материальные - 25681,31 руб.)								
№	Наименование операции	Время в	Специалист	Стоимость работ, Ресурс	Кол-во	Ед.изм.	Стоимость ресурс	Расх.коэф
1	Прием АД	0,50	Материально-отв	75,00				
2	Разборка АД	3,50	Электрослесарь,	393,75				
3	Мойка вспомогательных элементов АД	0,00	Электрослесарь,	112,50				
4	Снятие подшипников ротора	0,45	Электрослесарь,	33,75				
5	Установка новых подшипников ротора	0,00	Электрослесарь,	112,50	Подшипник	2,00	шт., 3000,00	1,00
6	Дефектация статора	1,25	Инженер-техноло	250,00				
7	Разборка электрической схемы обмотки статс	0,50	Инженер-техноло	100,00				
8	Вырубка лобовых части обмотки статора	3,00	Электрослесарь,	1012,50				
9	Загрузка статора в печь	0,25	Электрослесарь,	28,13				
10	Отжиг статора	7,00	Инженер-техноло	0,00	Электроэнергия	20,00	кВт/час 490,00	1,00
11	Выгрузка статора из печи	0,25	Электрослесарь,	28,13				
12	Экструзия обмотки из пазов статора	0,15	Электрослесарь,	810,00				
13	Чистка, правка магнитной системы статора	0,10	Электрослесарь,	360,00				
14	Инструментальное исследование качества ма	0,00	Инженер-техноло	200,00				
15	Электромагнитный пересчет параметров обмк	6,00	Инженер-техноло	1200,00				
16	Составление технологической карты ремонта	0,50	Инженер-техноло	100,00				
17	Изготовление блока компенсирующих конде-	8,00	Электрослесарь,	600,00			9805,47	
				1 кВАр компенсат		кВАр	8775,55	1,00
				Устройство защит		А	536,14	1,00

Рисунок 13 – Смета ремонта и модернизации АД без оптимизации

EngineRM								
Файл Справочники Результаты вычислений								
Главное меню Смета Графики затрат								
Смета затрат на ремонт с модернизацией асинхронного двигателя:								
☐ Без оптимизации ☒ С оптимизацией (Минимизация себестоимости)								
Время ремонта: 50,2 часов. Стоимость ремонта: 36320,56 руб. (затраты трудовые - 10639,25 руб., материальные - 25681,31 руб.)								
№	Наименование операции	Время в	Специалист	Стоимость работ, Ресурс	Кол-во	Ед.изм.	Стоимость ресурс	Расх.коэф
1	Прием АД	0,50	Материально-отв	75,00				
2	Разборка АД	3,50	Электрослесарь,	393,75				
3	Мойка вспомогательных элементов АД	0,00	Электрослесарь,	112,50				
4	Снятие подшипников ротора	0,45	Электрослесарь,	33,75				
5	Установка новых подшипников ротора	0,00	Электрослесарь,	112,50	Подшипник	2,00	шт., 3000,00	1,00
6	Дефектация статора	1,25	Инженер-техноло	250,00				
7	Разборка электрической схемы обмотки статс	0,50	Инженер-техноло	100,00				
8	Вырубка лобовых части обмотки статора	3,00	Электрослесарь,	1012,50				
9	Загрузка статора в печь	0,25	Электрослесарь,	28,13				
10	Отжиг статора	7,00	Инженер-техноло	0,00	Электроэнергия	20,00	кВт/час 490,00	1,00
11	Выгрузка статора из печи	0,25	Электрослесарь,	28,13				
12	Экструзия обмотки из пазов статора	0,15	Электрослесарь,	810,00				
13	Чистка, правка магнитной системы статора	0,10	Электрослесарь,	360,00				
14	Инструментальное исследование качества ма	0,00	Инженер-техноло	200,00				
15	Электромагнитный пересчет параметров обмк	6,00	Инженер-техноло	1200,00				
16	Составление технологической карты ремонта	0,50	Инженер-техноло	100,00				
17	Изготовление блока компенсирующих конде-	8,00	Электрослесарь,	600,00			9805,47	
				1 кВАр компенсат		кВАр	8775,55	1,00
				Устройство защит		А	536,14	1,00

Рисунок 14 – Смета ремонта и модернизации АД с оптимизацией

Для наглядного представления стоимости каждой технологической операции на проведения ремонта в программе на вкладке «Графики затрат» отображается график с выделением трудовых и материальных затрат. Пользователь может выбрать три типа графика – это трудовые и

материальные затраты, трудовые затраты, материальные затраты. На рисунке 15 представлен график на проведение ремонта АД с разделением трудовых и материальных затрат. Оранжевым цветом выделены трудовые затраты, синим – материальные. По оси абсцисс отложены технологические операции, на оси ординат – стоимость операции в рублях.

На рисунке 16 изображен график стоимости трудовых затрат при выполнении операции ремонта с модернизацией асинхронного двигателя типа «5А200L4Y3». В программе предусмотрена возможность подписать данные для наглядного представления стоимости технологической операции, рисунок 39.

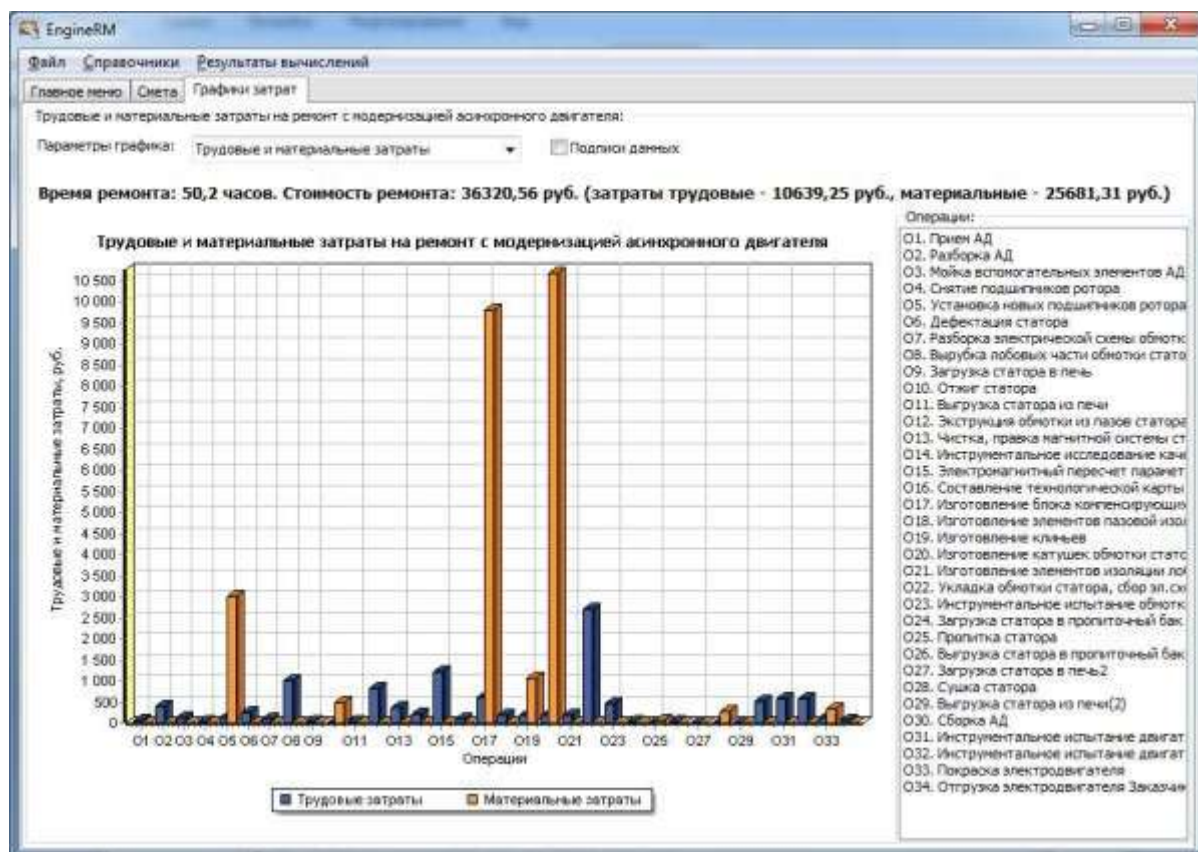


Рисунок 15 – График трудовых и материальных затрат

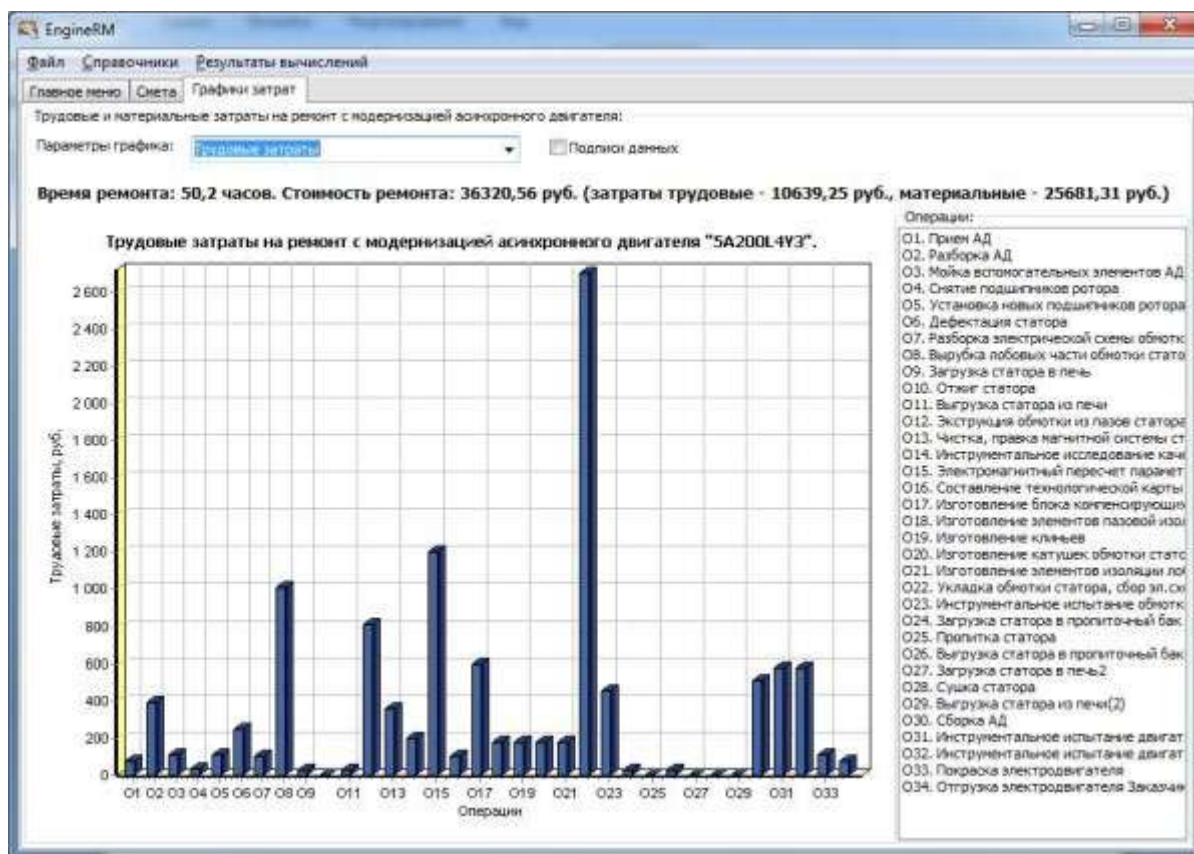


Рисунок 16 – График трудовых затрат

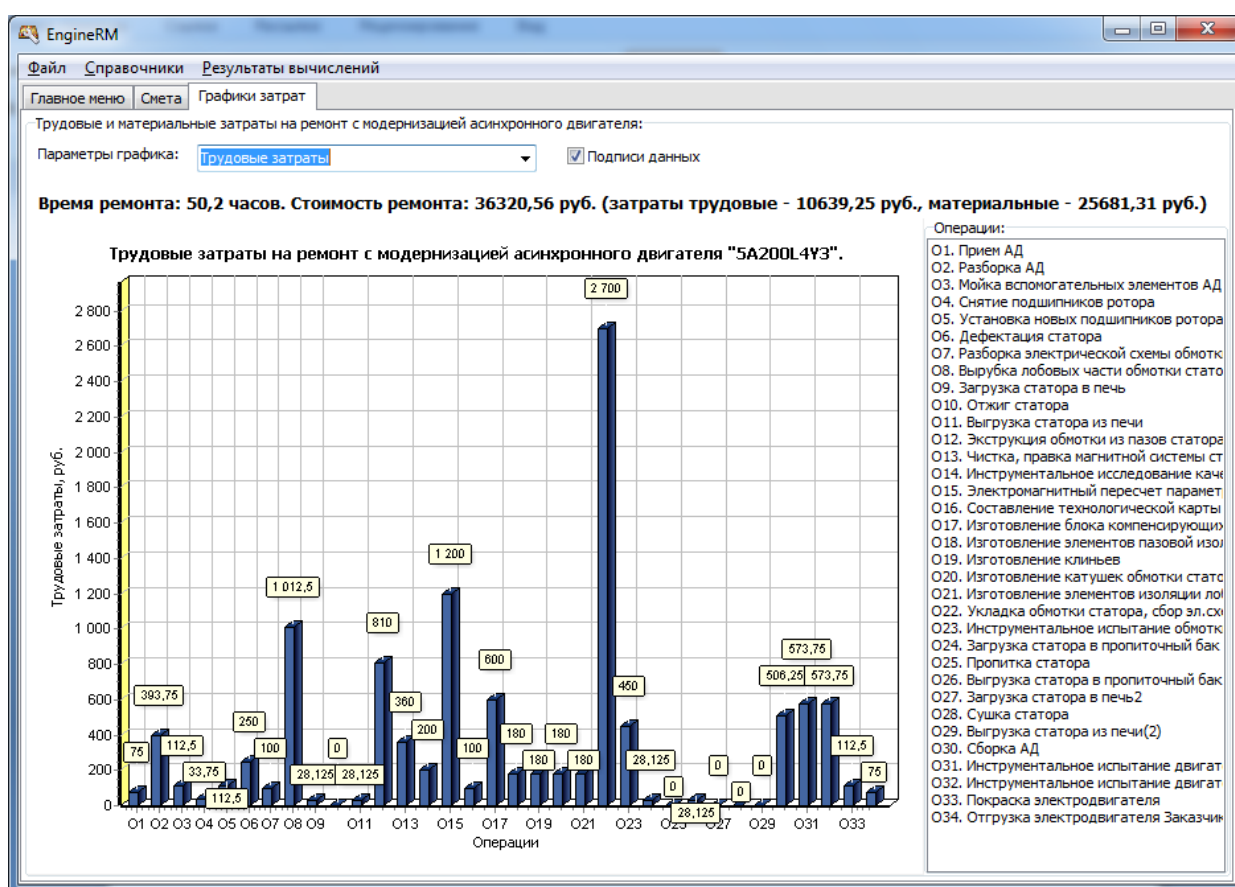


Рисунок 17 – График трудовых затрат

The screenshot shows the 'EngineM' application window. The 'Справочники' (Reference) menu is open, listing: Двигатели, Специалисты, Основные операции ремонта, Ресурсы, and Расходные коэффициенты. The main window displays input parameters for a motor (SA200L-4Y3) and repair type (Ремонт с модернизацией). It shows a table of motor characteristics and a comparison of costs before and after optimization.

Характеристика	Значение
Тип двигателя	SA200L-4Y3
Мощность двигателя, кВт	45,00
Номинальная частота вращения, об/мин	2940
Номинальная частота тока, Гц	50
Номинальный cos φ, о.е.	0,900
КПД, о.е.	93,400
Вес двигателя, кг	310
Номинальное фазное напряжение, В	0,81
Номинальный ток, А	82
Схема соединения обмотки	Звезда

Результаты до оптимизации		Результаты после оптимизации (Минимизация себестоимости)	
Трудовые затраты, руб.	11183,75	Трудовые затраты, руб.	10639,25
Материальные затраты, руб.	25681,31	Материальные затраты, руб.	25681,31
Общая стоимость, руб.	36865,06	Общая стоимость, руб.	36320,56

Рисунок 18 – Справочные таблицы

Для просмотра данных необходимо выбрать вкладку «Справочник», рисунок 18. Пользователю предлагается на выбор 5 справочных таблиц: двигатели, где хранятся параметры АД, специалисты, основные операции ремонта, ресурсы и расходный коэффициент.

На рисунке 19 представлено окно о программе.



Рисунок 19 – Окно о программе

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при

этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

Раздел 3. Разработка документации информационных систем

Практическое занятие № 9 **Изучение средств автоматизированного документирования**

Цель: изучить средства автоматизированного документирования

Выполнив работу, Вы будете:

уметь:

- У9. работать с инструментальными средствами обработки информации.

Материальное обеспечение:

Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства

Задание:

1. Произвести сравнительный анализ средств документирования

Порядок выполнения работы:

1. Изучить дополнительные источники и ответить на вопрос, что такое метод и способ документирования?
2. Сказать, что относится к традиционным способам документирования.
3. Сказать, что относится к технологичным способам документирования.
4. Какой программное обеспечение для автоматизированного документирования вы знаете?

Ход работы:

1. Выбрать дополнительные источники литературы.
2. Произвести анализ по обработке информации и ответить на 4 предложенных вопроса.
3. Результат выполнения лабораторного занятия оформить в виде отчета.

Форма представления результата:

Оформить отчет по выполненным заданиям.

Критерии оценки:

Оценка «отлично» выставляется: вся работа выполнена безошибочно и нет исправлений;

Оценка «хорошо» выставляется: допущены 1-2 вычислительные ошибки.

Оценка «удовлетворительно» выставляется: допущены ошибки в ходе решения задачи при правильном выполнении всех остальных заданий или допущены 3-4 вычислительные ошибки, при этом ход анализа должен быть верным.

Оценка «неудовлетворительно» выставляется: допущены ошибки в ходе решения задачи и хотя бы одна вычислительная ошибка или при анализе и примеров допущено более 5 вычислительных ошибок.

СОДЕРЖАНИЕ

1 Введение	4
2 Методические указания.....	6
Практическое занятие № 1 Знакомство со средой программирования.	6
Практическое занятие № 2 Составление программ линейной структуры.....	8
Практическое занятие № 3 Составление программ разветвляющейся структуры.	11
Практическое занятие № 4 Составление программ циклической структуры	14
Практическое занятие № 5 Обработка одномерных массивов	16
Практическое занятие № 6 Обработка двумерных массивов	18
Практическое занятие № 7 Работа со строками	20
Практическое занятие № 8 Работа с данными типа множество.	22
Практическое занятие № 9 Файлы последовательного доступа.....	24
Практическое занятие № 10 Типизированные файлы. Нетипизированные файлы.	26
Практическое занятие № 11 Организация процедур.	28
Практическое занятие № 12 Организация функций	30
Практическое занятие № 13 Применение рекурсивных функций.....	32
Практическое занятие № 14 Программирование модуля.....	34
Практическое занятие № 15 Создание библиотеки подпрограмм	36
Практическое занятие № 16 Использование указателей для организации связанных списков.....	38
Практическое занятие № 17 Изучение интегрированной среды разработчика.	40
Практическое занятие № 18 Создание проекта с использованием компонентов для работы с текстом.	43
Практическое занятие № 19 Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени	45
Практическое занятие № 20 События компонентов (элементов управления), их сущность и назначение	47
Практическое занятие № 21 Создание процедур на основе событий	49
Практическое занятие № 22 Создание проекта с использованием кнопочных компонентов.	51
Практическое занятие № 23 Создание проекта с использованием компонентов стандартных диалогов и системы меню.....	53
Практическое занятие № 24 Разработка функциональной схемы работы приложения.....	55
Практическое занятие № 25 Создание процедур обработки событий. Компиляция и запуск приложения.....	57
Практическое занятие № 26 Разработка оконного приложения с несколькими формами	59
Практическое занятие № 27 Разработка игрового приложения.	61
Практическое занятие № 28 Разработка интерфейса приложения.....	63
Практическое занятие № 29 Тестирование, отладка приложения.	65
Практическое занятие № 30 Классы ООП: виды, назначение, свойства, методы, события	68
Практическое занятие № 31 Объявления класса.....	70
Практическое занятие № 32 Создание наследованного класса	72
Практическое занятие № 33 Программирование приложений.....	74
Практическое занятие № 34 Перегрузка методов.....	77

1 ВВЕДЕНИЕ

Важную часть теоретической и профессиональной практической подготовки обучающихся составляют лабораторные занятия.

Состав и содержание лабораторных занятий направлены на реализацию Федерального государственного образовательного стандарта среднего профессионального образования.

Ведущей дидактической целью лабораторных занятий является формирование профессиональных практических умений (умений выполнять определенные действия, операции, необходимые в последующем в профессиональной деятельности), необходимых в последующей учебной деятельности.

Ведущей дидактической целью лабораторных занятий является экспериментальное подтверждение и проверка существенных теоретических положений (законов, зависимостей).

В соответствии с рабочей программой учебной дисциплины «Основы алгоритмизации и программирования» предусмотрено проведение лабораторных занятий.

В результате их выполнения, обучающийся должен:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы;

Содержание практических и лабораторных занятий ориентировано на подготовку обучающихся к освоению профессионального модуля программы подготовки специалистов среднего звена по специальности и овладению **профессиональными компетенциями:**

ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием.

ПК 9.2. Разрабатывать веб-приложение в соответствии с техническим заданием.

ПК 9.3. Разрабатывать интерфейс пользователя веб-приложений в соответствии с техническим заданием.

А также формированию **общих компетенций:**

ОК 01 - Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам;

ОК 02 - Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

ОК 04 - Эффективно взаимодействовать и работать в коллективе и команде.

ОК 05 - Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.

ОК 09 - Пользоваться профессиональной документацией на государственном и иностранном языках.

Выполнение обучающихся практических и/или лабораторных работ по учебной дисциплине «Основы алгоритмизации и программирования» направлено на:

- *обобщение, систематизацию, углубление, закрепление, развитие и детализацию полученных теоретических знаний по конкретным темам учебной дисциплины;*
- *формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;*
- *развитие интеллектуальных умений у будущих специалистов: аналитических, проективных, конструктивных и др.;*
- *выработку при решении поставленных задач профессионально значимых качеств, таких*

как самостоятельность, ответственность, точность, творческая инициатива.

Лабораторные занятия проводятся в рамках соответствующей темы, после освоения дидактических единиц, которые обеспечивают наличие знаний, необходимых для ее выполнения.

2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Тема 1.2. Типы данных

Практическое занятие № 1 Знакомство со средой программирования.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Какие операции определены над переменными булевого типа?
2. Записать на языке C#

$$y = \frac{\sqrt{ax + bx + c^2}}{15x - \sin x}$$

Дать протокол программы.

3. Какого типа будет результат деления 15 на 4?
4. Какие из приведенных ниже операторов присвоения являются правильными?
 - а. $x := I + J - B$
 - б. $I := I + K / J$

если известно, что I,J,K: integer; x,B: real?

5. Написать программу вычисления выражения

$$5.23 + 7.6^2 + \sin \frac{\pi}{7} \\ \sin \frac{2\pi}{7} + 3.1$$

Дать протокол программы.

Порядок выполнения работы:

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 2 Составление программ линейной структуры.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

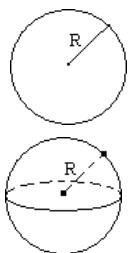
Calculate Linux Desktop

Задание:

1. Формулировка задачи

Составить алгоритм вычисления длины окружности, площади круга, площади сферы и объема шара по заданному радиусу окружности.

2. Математическая постановка задачи



Для расчета перечисленных характеристик воспользуемся формулами:

длина окружности – $L = 2\pi R$;

площадь круга – $S_{кр} = \pi R^2$;

площадь сферы – $S_{сф} = 4\pi R^2$;

объем шара – $V = \frac{4}{3}\pi R^3$,

где π – число Пи, математическая константа, которая выражает отношение длины окружности к её диаметру

$\pi \approx 3,141592653589793238462643...$, R – радиус окружности.

3. Выбор переменных программы

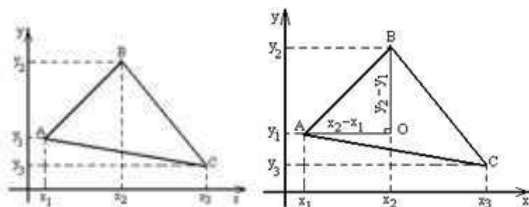
Из приведенного выше решения определяем следующие переменные:

исходные данные – радиус окружности (R);

справочные данные – число π (Pi);
результат – длина окружности (L), площадь круга ($S_{кр}$), площадь сферы ($S_{сф}$) и объем шара (V).

1. Формулировка задачи

Треугольник задается координатами своих вершин на плоскости: $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Требуется составить алгоритм программы, которая вычисляет площадь треугольника ABC .



2. Математическая постановка задачи

Для решения задачи можно использовать формулу Герона:

$S = p \cdot p - A \cdot p - B \cdot (p - C)$, где p – полупериметр. Для вычисления длины сторон по координатам вершин необходимо воспользоваться формулой $AB = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, где AB – длина стороны треугольника; (x_1, y_1) , (x_2, y_2) – координаты вершин A и B .

3. Выбор переменных программы

Из приведенного выше решения определяем следующие переменные:
исходные данные – координаты вершин на плоскости (x_1, y_1) , (x_2, y_2) , (x_3, y_3) ;
промежуточные данные – длины сторон (A , B , C), p – полупериметр.
результат – площадь треугольника (S).

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 3 Составление программ разветвляющейся структуры.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

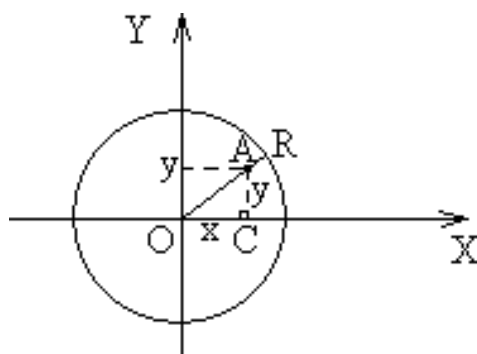
MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Составить алгоритм решения задачи, который определяет по введенным координатам, попадает ли заданная точка в окружность с центром в точке $O(0;0)$ и заданным радиусом R .
2. Математическая постановка задачи



Заданная точка имеет координаты x, y . Рассмотрим треугольник AOC . $\angle OCA = 90^\circ$, $OC = x$, $AC = y$, следовательно, по теореме Пифагора:
 AO^2 (гипотенуза) $= OC^2 + AC^2$ ($AO^2 = x^2 + y^2$).

Поэтому условие принадлежности точки окружности можно записать в виде: $x^2 + y^2 \leq R^2$.

3. Выбор переменных программы

Из приведенного выше решения определяем следующие переменные:

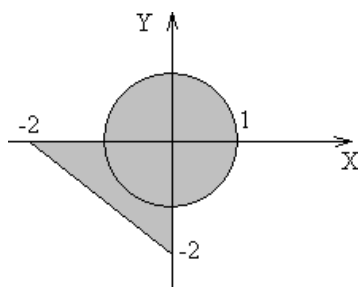
исходные данные – радиус окружности (R) и координаты точки (x и y);

результат – сообщение «В окружности» или «Вне окружности».

Так как радиус окружности и координаты точки могут принимать любые значения (2; 2,5; 3,75), все переменные для данной задачи определены как действительные числа.

1. Формулировка задачи

Составить алгоритм решения задачи, который по введенным координатам точки определяет, попадает ли эта точка в заштрихованную область.



2. Математическая постановка задачи

Запишем условия попадания точки в область в виде формул. Область можно описать как круг, пересекающийся с треугольником. Точка может попадать либо в круг, либо в треугольник, либо в их общую часть. Условие попадание точки в круг запишем как $x^2 + y^2 \leq 1^2$, т.к. радиус окружности в соответствии с графической интерпретацией равен 1.

Так как треугольник расположен в третьей четверти координатной плоскости, следовательно, для него выполняется условие $x \leq 0$ и $y \leq 0$. Выведем уравнение прямой линии, ограничивающей треугольник снизу.

Известно, что прямая линия проходит через две точки: $(-2; 0)$ и $(0; -2)$. Требуется составить уравнение прямой линии.

Общий вид уравнения прямой $y = kx + b$, где k, b – фиксированные числа.

Искомая прямая $y = kx + b$ с пока неизвестными коэффициентами k, b , проходит через точки $(-2; 0)$ и $(0; -2)$, а, значит, выполняются равенства $0 = k \cdot (-2) + b$ и $-2 = k \cdot 0 + b$, что можно записать в виде системы:

$$\begin{aligned} 0 &= k \cdot (-2) + b \\ -2 &= k \cdot 0 + b \end{aligned} \quad \text{или} \quad \begin{aligned} 0 &= k \cdot (-2) + b \\ -2 &= b \end{aligned}$$

Решив систему относительно неизвестных k и b , мы найдем уравнение прямой.

Подставим значение $b = -2$ в уравнение $0 = k \cdot (-2) + b$:

$$-2k = -b$$

$$-2k = -(-2)$$

$$-2k = 2$$

$$k = -1$$

Таким образом, искомое уравнение прямой имеет вид $y = (-1) \cdot x - 2$. Так как заштрихованная область расположена выше прямой линии, ограничивающей треугольник снизу, то попадание

$$x \leq 0$$

точки в треугольник запишем как $y \leq 0$. Следовательно, заштрихованная область

$$y \geq -x - 2$$

определена следующими условиями $x^2 + y^2 \leq 1^2$ или $x \leq 0$, $y \leq 0$.

$$y \geq -x - 2$$

3. Выбор переменных программы

Из приведенного выше решения определяем следующие переменные:

исходные данные – координаты точки (x и y);

результат – сообщение «Принадлежит» или «Не принадлежит».

Так как координаты точки могут принимать любые значения (2; 2,5; 3,75), все переменные определяем как действительные числа.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие №4 Составление программ циклической структуры

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи

Составить алгоритм расчета факториала $N!$ с использованием различных видов цикла.

2. Математическая постановка задачи

Факториал числа N (обозначается $N!$) – произведение всех натуральных чисел до N включительно:

$$N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$$

По определению полагают $0! = 1$. Факториал определён только для целых неотрицательных чисел.

3. Выбор переменных программы

Из приведенного выше решения определяем следующие переменные:

исходные данные – значение N ;

результат – значение *factorial*.

Выбор переменных: аргумент и значение функции могут принимать целые неотрицательные значения.

1. Формулировка задачи

Составить алгоритм вычисления суммы ряда:

$$y = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + \frac{(-1)^{n-1} x^n}{n} \text{ с точностью } e=0.0001 \text{ с использованием цикла с}$$

постусловием и предусловием.

2. Математическая постановка задачи

Ряд представляет собой сумму слагаемых, в котором перед каждым слагаемым чередуется знак (перед первым слагаемым знак «+», перед вторым – «-», перед третьим – «+», перед четвертым – «-», и т.д.). Поэтому для его учета в алгоритм накопления суммы вводим переменную, отвечающую за знак – z.

При накоплении суммы, начальное значение суммы обнуляем, т.е. S=0. Первое слагаемое можно представить в виде $\frac{x^1}{1}$, т.е. начальное значение степени x и знаменатель равны 1. Шаг изменения степени и знаменателя совпадает и равен 1. Точность вычисления определяется значением слагаемого.

3. Выбор переменных программы

Из приведенного выше описания определяем следующие переменные:

- исходные данные – значение x;
- начальное значение суммы S равно 0 (S=0);
- начальное значение степени x и знаменатель равны 1 (i=1);
- знак перед первым слагаемым – «+» (z=1);
- результат – значение S.

Порядок выполнения работы:

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата:

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном

сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 5 Обработка одномерных массивов.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

Заполнение массива с помощью генератора случайных чисел

Для генерации чисел в диапазоне $[A, B]$ можно использовать следующее выражение:
 $random(B-A+1)+A$.

Например, выражение $A[i]=random(100)$ генерирует числа от 0 до 99, при этом 100 не входит в диапазон.

Записать выражение для генерации чисел в диапазоне $[-15; 38]$:

$A[i]=random(38-(-15)+1)+(-15)$, т.е. $A[i]=random(54)-15$.

1. Формулировка задачи

Заполнить массив n целыми случайными числами в диапазоне $[-10; 15]$. Составить алгоритм вычисления суммы четных чисел.

2. Математическая постановка задачи

При накоплении суммы, начальное значение суммы обнуляем, т.е. $\text{Sum}=0$.

Выражение для генерации чисел в диапазоне $[-10; 15]$:

$A[i]=\text{random}(15-(-10)+1)+(-10)$, т.е. запишем в следующем виде $A[i]=\text{random}(26)-10$.

Для четного значения элемента выполняется условие $A[i] \bmod 2=0$.

3. Выбор переменных программы

Из приведенного выше описания определяем следующие переменные:

исходные данные – количество элементов массива n ;

начальное значение суммы равно 0 ($\text{Sum}=0$);

результат – значение накопленной суммы Sum .

Задача

1. Формулировка задачи

Заполнить массив n целыми случайными числами в диапазоне $[-5; 10]$. Составить алгоритм вычисления произведения положительных элементов.

2. Математическая постановка задачи

При накоплении произведения, начальное значение произведения равно 1, т.е. $\text{pr}=1$.

Выражение для генерации чисел в диапазоне $[-5; 10]$:

$A[i]=\text{random}(10-(-5)+1)+(-5)$, т.е. запишем в следующем виде $A[i]=\text{random}(16)-5$.

Для положительного значения элемента выполняется условие $A[i] > 0$.

3. Выбор переменных программы

Из приведенного выше описания определяем следующие переменные:

исходные данные – количество элементов массива n ;

начальное значение произведения равно 1 ($\text{pr}=1$);

результат – значение произведения pr .

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания

выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 6 Обработка двумерных массивов.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Составить алгоритм и написать программу:

Вычислить сумму и число положительных элементов матрицы $A[N, N]$, находящихся над главной диагональю.

1. Формулировка задачи:

Составить алгоритм и написать программу:

Дана матрица $B[N, M]$.. Найти в каждой строке матрицы максимальный и минимальный элементы и поменять их местами с первым и последним элементом строки соответственно.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 7 Работа со строками.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)
MS Office 2007
7 Zip
MS Visual Studio (подписка Imagine Premium)
Visual Studio Code
Calculate Linux Desktop

Задание:

1. Формулировка задачи

- Дана строка, заканчивающаяся точкой. Подсчитать, сколько слов в строке.
- *Лишние пробелы.* Дана строка, состоящая из слов, разделенных пробелами. Напишите программу, удаляющую лишние пробелы. Пробел считается лишним, если он: стоит в начале строки.

1. Формулировка задачи:

- Дана строка. Подсчитать, сколько в ней букв R, K, T.
- *Лишние пробелы.* Дана строка, состоящая из слов, разделенных пробелами. Напишите программу, удаляющую лишние пробелы. Пробел считается лишним, если он: следует за пробелом.

2. Выбор переменных программы

Порядок выполнения работы

- Составить математическую модель задачи.

- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 8 Работа с данными типа множество.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Дана непустая последовательность символов. Требуется построить и напечатать множество, элементами которого являются символы, встречающиеся в последовательности индивидуального варианта:

Вариант Последовательность символов

1 Буквы от 'A' до 'F' и от 'X' до 'Z';

2 Цифры от '5' до '9' и знаки арифметических операций;

4 Буквы от 'T' до 'X' и цифры от '1' до '4';

5 Знаки препинания и операций отношения;

6 Знаки арифметических операций и буквы от 'E' до 'N';

7 Буквы от 'A' до 'Z' и знаки препинания;

8 Знаки операций отношения;

9 Цифры от '0' до '9';

10 Знаки арифметических операций и операций отношения.

2. При выполнении задания необходимо:

- а) ознакомиться с конечным и упорядоченным множеством символов, определенным на используемой для выполнения задания ЭВМ;
- б) составить программу для конкретного варианта, работающую для произвольного набора символов.
- с) входная строка символов может быть длиннее строки экрана терминала, при этом программа работает не с функцией EOLN, а с признаком конца строки, который задается программистом.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 9 Файлы последовательного доступа.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Записать в файл прямого доступа ряд вещественных чисел. Найти наибольшее из значений модулей компонентов с нечетными номерами.

1. Формулировка задачи:

Записать в файл последовательного доступа произвольных натуральных чисел. Переписать в другой файл последовательного доступа те элементы, которые кратны K .

Вывести полученный файл на печать.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1. Операторы языка программирования

Практическое занятие № 10 Типизированные файлы. Нетипизированные файлы.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)
MS Office 2007
7 Zip
MS Visual Studio (подписка Imagine Premium)
Visual Studio Code
Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Дана последовательность целых чисел, оканчивающаяся нулем. Записать все числа последовательности в типизированный файл.

В конец существующего типизированного файла записать:

- а) число 0;
- б) фразу «До свидания!».

1. Формулировка задачи:

Создать типизированный файл, элементами которого являются двенадцать первых членов последовательности Фибоначчи (последовательности, в которой первые два члена равны 1, а каждый следующий равен сумме двух предыдущих).

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;

- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 3.1. Процедуры и функции

Практическое занятие № 11 Организация процедур.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)
MS Office 2007
7 Zip
MS Visual Studio (подписка Imagine Premium)
Visual Studio Code
Calculate Linux Desktop

Задание:

1. Формулировка задачи:

- Разработать программу, используя модульный подход.
- Сформировать по одному или несколько заголовочных модулей и файлов реализации, которые будут содержать разработанные функции.
- Написать главную функцию `main`, используя правила структурного подхода: она должна содержать в основном только вызов функций, все вычисления должны быть внутри разработанных функций.
- Выполнить отладку и компиляцию программы, получить исполняемый файл.
- Выполнить тестирование программы несколькими наборами входных данных.
- Составить диаграмму модулей, дерево вызова функций и спецификации функций

Даны целые числа $a_1, \dots, a_n, b_1, \dots, b_m, k$.

Если в последовательности $a_1, \dots, a_n$ нет ни одного члена со значением k , то первый по порядку член этой последовательности, не меньший всех остальных членов, заменить на значение k .

По такому же правилу преобразовать последовательность $b_1, \dots, b_m$ применительно к 10.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 3.1. Процедуры и функции

Практическое занятие № 12 Организация функций.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Написать функцию, выводящую в порядке возрастания элементы одномерного массива. В главной программе вызвать функцию для двух разных массивов.

1. Формулировка задачи:

Написать функцию вычисления произведения прямоугольной матрицы A размера $k \times m$ на прямоугольную матрицу B размера $m \times n$. В главной программе обратиться к этой функции.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с

входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 3.1. Процедуры и функции

Практическое занятие № 13 Применение рекурсивных функций.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)
MS Office 2007
7 Zip
MS Visual Studio (подписка Imagine Premium)
Visual Studio Code
Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Ввести с клавиатуры целое число N. Используя рекурсию, распечатать сначала последовательность, состоящую из N букв 'A', а затем из N букв 'B'.

1. Формулировка задачи:

Напечатать в обратном порядке последовательность чисел, признаком конца которой является 0.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные

- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 3.3. Модульное программирование

Практическое занятие № 14 Программирование модуля.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Средствами модульного программирования решить задачу обработки одномерного числового массива.

Найти сумму элементов до первого положительного и произведение остальных элементов.

1. Формулировка задачи:

Средствами модульного программирования решить задачу обработки одномерного числового массива.

Найти количество элементов, стоящих на нечётных местах и меньших заданного числа.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 3.3. Модульное программирование

Практическое занятие № 15 Создание библиотеки подпрограмм.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1.Формулировка задачи:

Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над обыкновенными дробями вида P/Q (P - целое, Q - натуральное):

- сложение;
- вычитание;
- умножение;
- деление;
- сокращение дроби;
- возведение дроби в степень N (N - натуральное);
- функции, реализующие операции отношения (равно, не равно, больше или равно, меньше или равно, больше, меньше).

Используя этот модуль, решить задачу:

Дан массив A - массив обыкновенных дробей. Найти сумму всех дробей, ответ представить в виде несократимой дроби. Вычислить среднее арифметическое всех дробей, ответ представить в виде несократимой дроби.

1. Формулировка задачи:

Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над обыкновенными дробями вида P/Q (P - целое, Q - натуральное):

- сложение;
- вычитание;
- умножение;
- деление;
- сокращение дроби;
- возведение дроби в степень N (N - натуральное);
- функции, реализующие операции отношения (равно, не равно, больше или равно, меньше или равно, больше, меньше).

Используя этот модуль, решить задачу:

Дан массив A - массив обыкновенных дробей. Отсортировать его в порядке возрастания.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 4.1 Указатели.

Практическое занятие № 16 Использование указателей для организации связанных списков.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Сформировать односвязный список из 5 записей, удалить записи, удовлетворяющие некоторому условию, и вывести список на экран.

Запись содержит марку магнитофона, качество, цену. Удалить из списка записи с ценой ниже 800 руб.

1. Формулировка задачи:

Сформировать односвязный список из 5 записей, удалить записи, удовлетворяющие некоторому условию, и вывести список на экран.

Запись содержит фамилию и 4 оценки. Удалить из списка записи с неудовлетворительными оценками

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.

- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 4.1 Указатели.

Практическое занятие № 17 Изучение интегрированной среды разработчика.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

Формулировка задачи:

Изучить алгоритм решения задачи определения траектории полета снаряда, запущенного под углом к горизонту.

Уравнения, описывающие движение снаряда, т.е. зависимость его координат x и y от времени t , прошедшего с момента выстрела, выглядят следующим образом:

$$\begin{aligned}x(t) &= x_0 + V_{0x}t; \\ y(t) &= y_0 + V_{0y}t - \frac{gt^2}{2}; \\ V_{0x} &= V_0 \cos \alpha; \\ V_{0y} &= V_0 \sin \alpha.\end{aligned}$$

Здесь введены следующие обозначения:

x_0, y_0 — координаты начальной точки траектории полета снаряда, м;

V_0 — начальная скорость снаряда, м/с;

α — угол наклона начального участка траектории полета снаряда, рад;

V_{0x}, V_{0y} — проекции вектора начальной скорости V_0 на координатные оси x и y , соответственно, м/с;

g — ускорение свободного падения, $g = 9,81 \text{ м/с}^2$.

Будем считать, что необходимо определить положение (координаты) снаряда в моменты времени $0 \text{ с}, \Delta t, 2\Delta t, \dots, (n-1)\Delta t$, причем

$$\Delta t = \frac{T}{n-1},$$

где T — время полета снаряда, с,

n — число точек, в которых рассчитываются координаты траектории полета снаряда.

Величина T определяется по формуле

$$T = \frac{2V_{0y}}{g}.$$

Исходя из записанных выше уравнений, алгоритм решения задачи состоит из следующих действий:

1. Ввести начальные значения скорости снаряда V_0 , угла наклона начального участка траектории α и число точек n .
2. Вычислить значение V_{0x}
3. Вычислить значение V_{0y}
4. Присвоить $g = 9,81 \text{ м/с}^2$.
5. Вычислить T
6. Вычислить Δt
7. Присвоить $i = 1$.
8. Присвоить $t = 0$.
9. Вычислить $x(t)$
10. Вычислить $y(t)$
11. Вывести координаты снаряда.
12. Присвоить $t = t + \Delta t$.
13. Присвоить $i = i + 1$.
14. Если $i \leq n$, то перейти к шагу 9, иначе остановить выполнение программы.

Написать программу, вычисляющую координаты траектории полета снаряда

Выполнить компиляцию программы.

Исправить синтаксические ошибки, если такие имеются, и повторить компиляцию программы.

Запустить программу и проанализировать результаты ее работы.

Добавить в окно просмотра *Watch* переменные t, T, n, i .

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 4.1 Указатели.

Практическое занятие № 18 Создание проекта с использованием компонентов для работы с текстом.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

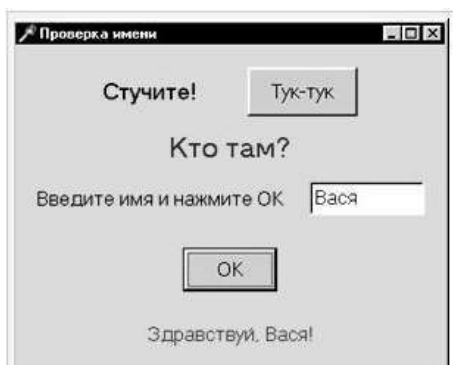
Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Компьютер должен запросить имя пользователя (пароль). Если пользователь дает правильный ответ (вводит знакомое имя, например, Вася), то компьютер приветствует его. В случае ввода любого другого слова, компьютер должен реагировать иначе, например, выводить запрос: "А где Вася?"



Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)

Практическое занятие № 19 Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Создайте приложение, при выполнении которого на экране появляется окно с тем же заголовком и с теми же компонентами, отображающими дату и время, что и на рисунке. Цвета частей календарей должны отличаться от стандартного цвета. Компоненты должны отображать текущую дату или текущее время. Компонент календаря должен быть установлен с текущим месяцем в середине этого компонента с использованием кнопок в его верхней части. При демонстрации выполнения приложения один из компонентов даты и времени должен быть раскрыт так же, как на рисунке



Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)

Практическое занятие № 20 События компонентов (элементов управления), их сущность и назначение

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. Прочтите в Приложении 1 описание свойств и описание элементов управления Form, Label, TextBox, Button. С помощью окна свойств установите первоначальный цвет формы, шрифт выводимых символов.

Задание

- Разместите на форме ряд кнопок (Button) напротив каждой поле ввода (TextBox) и одну метку (Label). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст в метке. Текст в метке берется из поля ввода напротив нажимаемой кнопки.
- Разместите на форме ряд кнопок (Button), и одно поле ввода (TextBox). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст на нажатой кнопке. Текст на кнопке берется из поля ввода.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)

Практическое занятие № 21 Создание процедур на основе событий

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Составить программу, отображающую графики функций $\sin(x)$ и $\cos(x)$ на интервале $[X_{\min}, X_{\max}]$. Предусмотреть возможность изменения разметки координатных осей, а так же шага построения таблицы. Прежде всего, следует поместить на форму сам элемент управления Chart. Он располагается в панели элементов в разделе Данные. Список графиков хранится в свойстве Series, который можно изменить, выбрав соответствующий пункт в окне свойств. Поскольку на одном поле требуется вывести два отдельных графика функций, нужно добавить еще один элемент. Оба элемента, и существующий и добавленный, нужно соответствующим образом настроить: изменить тип диаграммы ChartType на Spline. Здесь же можно изменить подписи к графикам с абстрактных Series1 и Series2 на $\sin(x)$ и $\cos(x)$ – за это отвечает свойство Legend. Наконец, с помощью свойства BorderWidth можно сделать линию графика по - толще, а затем поменять цвет линии с помощью свойства Color.

Задание

Постройте график функции для своего варианта. Таблицу данных получить путем изменения

параметра X с шагом dx . Добавьте второй график для произвольной функции.

1) $y = 10^{-2}bc / x + \cos \sqrt{a^3 x},$

$$x_0 = -1.5; x_k = 3.5; dx = 0.5;$$

$$a = -1.25; b = -1.5; c = 0.75;$$

2) $y = 1.2(a-b)^3 e^{x^2} + x,$

$$x_0 = -0.75; x_k = -1.5; dx = -0.05;$$

$$a = 1.5; b = 1.2;$$

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.2 Интегрированная среда разработчика.

Практическое занятие № 22 Создание проекта с использованием кнопочных компонентов.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1.Формулировка задачи:

Когда происходит какое – либо событие (например, событие Click при нажатии на кнопку), в обработчик этого события передаются до-полнительные сведения об этом событии в параметре e. Например, при щелчке кнопки мыши на объекте возникает событие MouseClick. Для этого события параметр e содержит целый ряд переменных, которые позволяют узнать информацию о нажатии:

- Button – какая кнопка была нажата;
- Clicks – сколько раз была нажата и отпущена кнопка мыши;
- Location – координаты точки, на которую указывал курсор в момент нажатия, в виде объекта классаPoint;
- X и Y – те же координаты в виде отдельных переменных.

Задание

Если в индивидуальном задании используется элемент Panel, измените его цвет, чтобы он визуально выделялся на форме. Если используется элемент Label, не забудьте присвоить ему какой – либо текст, иначе он не будет виден на форме.

-Разработать программу, динамически порождающую на окне кнопки. Левый верхний угол кнопки определяется местоположением курсора при щелчке. Вывести надпись на кнопке с координатами ее левого верхнего угла.

-Разработать программу, динамически порождающую на окне кнопки и поля ввода. Левый верхний угол элемента управления определяется местоположением курсора при щелчке. Кнопка порождается, если курсор находится в левой половине окна, в ином случае порождается поле ввода.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.2 Интегрированная среда разработчика.

Практическое занятие № 23 Создание проекта с использованием компонентов стандартных диалогов и системы меню.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Реализовать программно элемент управления «Список с прокруткой» для текстового режима. Работа со списком должна сводиться к вызову одной функции, в которую передаются координаты области на экране, которую занимает список, и массив строк, составляющих список. Функция самостоятельно организует интерфейс для выбора элемента.

Каждая строка представляет один элемент списка. В строках не должно быть символов возврата каретки и перевода строки. Элементы должны выводиться в заданном окне. Количество строк в списке может как превышать высоту окна, так и быть меньше ее. В первом случае необходимо предусмотреть прокрутку содержимого в окне, во втором – обеспечить корректный выбор только тех позиций в окне, которые содержат строки списка (другими словами, курсор не должен выходить за пределы и окна, и списка).

Текущий элемент подсвечивается каким-либо цветом, заданным разработчиком. Выбор элемента осуществляется нажатием клавиш «Up» (перемещение курсора вверх) и «Down» (перемещение курсора вниз).

Подтверждение выбора (и выход из функции) осуществляется при нажатии клавиш Enter, Esc, Left, Right, Tab, Shift+Tab. Каждой из этих клавиш должен соответствовать определенный код,

который также возвращается функцией выбора из списка. Этот код может использоваться вызывающей функцией для того, чтобы определить дальнейшее действие: переход к предыдущему (Shift+Tab, Left) или следующему (Tab, Right) элементу управления, выполнение основного действия программы (Enter), выход из программы (Esc).

Помимо указанного кода функция должна возвращать индекс выбранного элемента списка.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.2 Интегрированная среда разработчика.

Практическое занятие № 24 Разработка функциональной схемы работы приложения.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

Формулировка задачи:

Функциональная схема – это схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Для изображения функциональных схем используют специальные обозначения, установленные стандартом ГОСТ 19.701-90 (ИСО 5807-85).

Задание

Создать приложение Сортировка, в котором с клавиатуры вводятся элементы массива целых чисел, производится сортировка элементами различными методами, результаты сортировки выводятся на экран. Эскиз функциональной схемы приведен на рисунке 1.

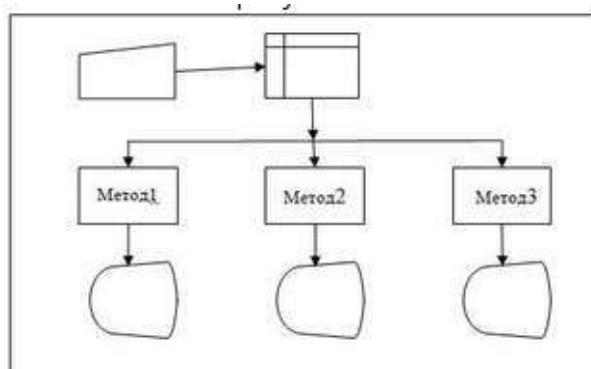


Рис. 1. Функциональная схема приложения

1. Разработать интерфейс и написать программные коды для событий кнопок.
2. Провести тестирование и отладку программы.
3. Нарисовать интерфейс программы со спецификацией и записать программные коды с комментариями в отчете по работе.
4. Записать несколько вариантов тестирования программы.
5. Провести тестирование исполняемого файла.

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.3. Визуальное событийно-управляемое программирование

Практическое занятие № 25 Создание процедур обработки событий. Компиляция и запуск приложения

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Написать функцию ввода текстовой строки с редактированием. При вызове функции ей передается длина поля ввода (максимальная длина отображаемой на экране части строки), а также координаты первого символа поля ввода на экране. Поле ввода однострочное, т.е. имеет высоту, равную 1 знакоместу.

Длина вводимой строки может превышать размеры поля ввода. Если в процессе ввода или перемещения курсора по строке текущая позиция ввода выходит за пределы поля, необходимо производить прокрутку строки в окне.

По усмотрению разработчика максимальная длина строки может либо передаваться в функцию, либо определяться объемом доступной динамической памяти. В первом случае буфер под строку резервирует вызывающая функция, во втором – функция ввода.

Функция ввода должна предоставлять следующие возможности по редактированию текста: перемещение курсора, ввод символа, удаление символа над курсором (клавиша Delete), удаление символа перед курсором (клавиша Backspace).

Если часть строки находится за пределами поля, желательно показывать это каким-либо образом (например, ставить выделенные цветом символы «<» и «>» в начале и конце поля ввода, если есть текст за пределами поля с соответствующей стороны).

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.4 Разработка оконного приложения

Практическое занятие № 26 Разработка оконного приложения с несколькими формами.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Разработать многооконное приложение.

- На главную форму приложения поместить компонент TAnimate. Компонент должен реализовать просмотр пользовательской анимации, путь которому прописывается в свойстве FileName.

- На главной форме реализовать прослушивание музыкальной композиции при помощи компонента TMediaPlayer.

- Поместить на форму главное меню, содержащее следующие пункты:

- задание 1

- задание 2

- выход

При выборе пункта меню «Задание 1», необходимо реализовать открытие формы «График» (в модальном режиме).

На форме «График» разместить 2 вкладки:

- «Таблица»- содержит таблицу, содержащее протабулированное значение функции $y=f(x)$ на отрезке $[a,b]$. Количество значений функций равно n . Шаг вычисляется по формуле $h=(b-a)/n$.

- «График» - содержит график функции $y=f(x)$, начерченный с помощью компонента TChart. При выборе пункта меню «Задание 2», необходимо открытие формы «Теория» (в немодальном режиме). На форме «Теория» расположить 2 компонента TreeView и WebBrowser. Узлами дерева TreeView являются темы теоретического материала. По щелчку названию темы необходимо осуществить загрузку соответствующей страницы html в компонент WebBrowser. При выборе пункта меню «Выход», необходимо закрыть приложение.

№	Функция	Интервал
1.	$Y=2\sin(x/3)$	$[-\pi/2; \pi/2]$
2.	$Y=2\cos(x/4)$	$[0; 3\pi/2]$
3.	$Y= \sin(x)+\cos(x) $	$[0; \pi]$
4.	$Y= \sin(x)-\cos(x) $	$[0; \pi]$
5.	$Y=2\sin(x)+3\cos(x)$	$[-\pi; \pi]$
6.	$Y=\sin(x)+\cos^2(x)$	$[-\pi; \pi]$
7.	$Y=2-\cos(x)$	$[0; 3\pi/2]$
8.	$Y=\sin(\sqrt{2}x)+\cos(x)$	$[0; 2\pi]$
9.	$Y=2\sin(2x)+1$	$[-\pi/2; \pi/2]$
10.	$Y=\sin(x)+\cos(x)-1$	$[-\pi; \pi]$

2. Создать внешний вид формы программы
3. Написать программные коды для созданной формы
4. Запустить и отладить программу

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.4 Разработка оконного приложения

Практическое занятие № 27 Разработка игрового приложения.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)
MS Office 2007
7 Zip
MS Visual Studio (подписка Imagine Premium)
Visual Studio Code
Calculate Linux Desktop

Задание:

1.Формулировка задачи:

Игра «Угадай карту». Дано пять карт, четыре короля и дама пик, перевернутые «рубашкой вверх». Цель игры: угадать, где находится дама пик за наименьшее число ходов. Щелчок по карте (ход) переворачивает ее. Добавить звуковое сопровождение. (3 балла).

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

1.Формулировка задачи:

Игра «Где яйцо». Даны три корзины. В одной из корзин спрятано яйцо. Нужно угадать в какой корзине яйцо. Щелчком мыши корзина переворачивается. Добавить звуковое сопровождение. (3 балла).

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;

- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.5 Этапы разработки приложений

Практическое занятие № 28 Разработка интерфейса приложения.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Для программы требуется разработать консольный интерфейс, интерфейс-меню и интерфейс со свободной навигацией. Результатом разработки является текстовое описание запросов, пунктов меню или компонентов интерфейса соответственно.

Разработать описание пользовательского интерфейса для программы «Будильник».

2. Исходными данными для программы являются:

- время, на которое установлен будильник;
- флажок, показывающий, включен будильник или выключен;
- период повторения будильника: однократное срабатывание, включение каждый день, включение раз в неделю.

3. Создать внешний вид формы программы

4. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;

- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.5 Этапы разработки приложений

Практическое занятие № 29 Тестирование, отладка приложения.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

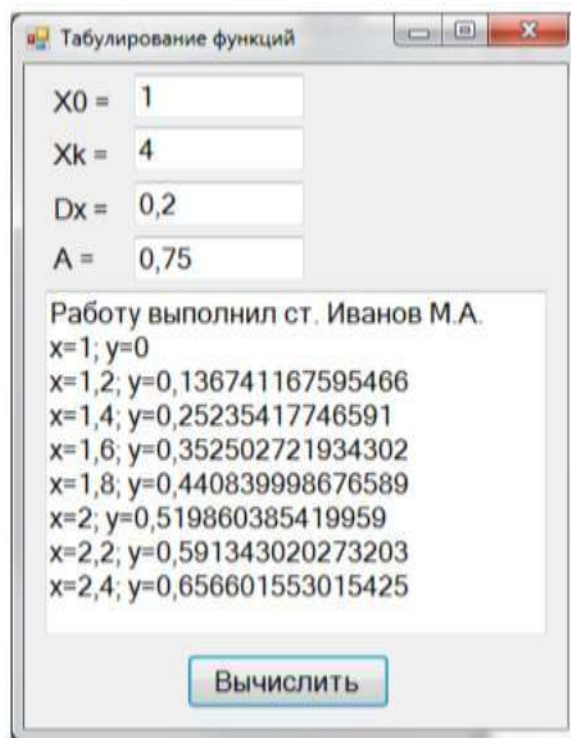
Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Вычислить и вывести на экран таблицу значений функции $y=a \cdot \ln(x)$ при x , изменяющемся от x_0 до x_k с шагом dx , a – константа. Панель диалога представлена на рисунке. После отладки программы следует проверить правильность работы программы с помощью контрольного примера (см. рис.). Установите точку останова на оператор перед циклом и запустите программу. После попадания на точку останова, выполните пошагово программу и проследите, как меняются все переменные в процессе выполнения.



Составьте программу табулирования функции $y(x)$, выведите на экран значения x и $y(x)$. Нужный вариант задания выберите из нижеприведенного списка по указанию преподавателя. Откорректируйте элементы управления в форме в соответствии со своим вариантом задания.

$$1) \quad y = 10^{-2}bc / x + \cos \sqrt{a^3 x},$$

$$x_0 = -1.5; x_k = 3.5; dx = 0.5;$$

$$a = -1.25; b = -1.5; c = 0.75;$$

$$2) \quad y = 1.2(a-b)^3 e^{x^2} + x,$$

$$x_0 = -0.75; x_k = -1.5; dx = -0.05;$$

$$a = 1.5; b = 1.2;$$

2. Создать внешний вид формы программы
3. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания

выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.6 Иерархия классов.

Практическое занятие № 30 Классы ООП: виды, назначение, свойства, методы, события.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и поворота на заданный угол.

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

Задача

1. Формулировка задачи:

Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.

- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.6 Иерархия классов.

Практическое занятие № 31 Объявления класса.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)
MS Office 2007
7 Zip
MS Visual Studio (подписка Imagine Premium)
Visual Studio Code
Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Определить класс список элементов. В определение класса включить два конструктора для определения списка по его размеру и путем копирования другого списка. Предусмотреть функции инверсии списка (123->321) и поиска подсписка в списке.

2. Создать внешний вид формы программы
3. Написать программные коды для созданной формы

1. Формулировка задачи:

Определить класс сортированный список элементов. В определение класса включить два конструктора для определения списка по его размеру и путем копирования другого списка. Предусмотреть функции добавления элемента и слияния двух сортированных списков.

2. Создать внешний вид формы программы
3. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;

- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.6 Иерархия классов.

Практическое занятие № 32 Создание наследованного класса.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

1. Формулировка задачи:

Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров, построения наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух

прямоугольников.

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.6 Иерархия классов.

Практическое занятие № 33 Программирование приложений.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)
MS Office 2007
7 Zip
MS Visual Studio (подписка Imagine Premium)
Visual Studio Code
Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Написать Windows-объектов в приложениях на языке C++ приложение, заголовок главного окна которого содержит Ф. И. О., группу и номер варианта. Программа должна динамически отображать информацию об изменении заданной характеристики компьютера в графической форме, используя компонент Chart или средства рисования GDI. График должен отображать значение заданной характеристики с автоматической прокруткой влево по мере появления новых данных (Ctrl+Shift+N приводит к тому же результату), в котором выбираем кроме столбчатых и круговых диаграмм). Промежуток времени между последовательными замерами значения характеристики, а также дополнительные параметры отображения графика в соответствии с заданием должны задаваться в диалоговом окне.

1).. Отображать график загрузки процессора (Ctrl+Shift+N приводит к тому же результату), в котором выбираем в диалоговом окне выбирать номер процессора)

2 Отображать текущий размер заданной папки (Ctrl+Shift+N приводит к тому же результату), в котором выбираем имя папки задается в диалоговом окне)

3 Отображать размер очереди печати заданного в диалоговом окне принтера.

4 Отображать количество запущенных в системе процессов (Ctrl+Shift+N приводит к тому же результату), в котором выбираем в диалоговом окне задавать шаблон имен отслеживаемых процессов).

5 Отображать процент загрузки процессора заданным в диалоговом окне процессом.

6 Отображать объем занимаемой заданным в диалоговом окне процессом виртуальной памяти

7 Отображать количество отправленных пакетов через указанный в диалоговом окне сетевой интерфейс.

8 Отображать частоту обращений на чтение/запись к диску, выбранному в диалоговом окне

9 Отображать количество процессов с различными значениями приоритетов в виде столбчатой диаграммы.

10 Отображать количество работающих или остановленных служб (Ctrl+Shift+N приводит к тому же результату), в котором выбираем переключение типов графиков реализовать в диалоговом окне)

11 Отображать размер файла подкачки и свободной физической памяти (Ctrl+Shift+N приводит к тому же результату), в котором выбираем переключение типов графиков реализовать в диалоговом окне).

12 Отображать количество потоков, запущенных в различных процессах, в виде столбчатой диаграммы (Ctrl+Shift+N приводит к тому же результату), в котором выбираем количество столбцов-объектов в приложениях на языке C++ процессов задавать в диалоговом окне).

13 Отображать объем свободной памяти на логических дисках в виде круговой диаграммы (Ctrl+Shift+N приводит к тому же результату), в котором выбираем в диалоговом окне предоставить возможность включать или отключать в диаграмму информации о сменных носителях).

14 Отображать в виде столбчатой диаграммы объем используемой памяти у 10 процессов с наибольшим значением этого параметра.

15 Отображать динамику использования оперативной памяти в виде круговой диаграммы с секторами, соответствующими: 5 наиболее активно использующих память процессов, объему памяти, используемым стальными процессами, а также объему свободной памяти.

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.
- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые

умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 5.6 Иерархия классов.

Практическое занятие № 34 Перегрузка методов.

Цель работы:

- Научиться применять базовые понятия теории приближенных методов решения задач на ЭВМ.
- Сформировать представление об алгоритмах решения прикладных задач, научиться написанию и отладки программного кода.
- Выбрать и обосновать наиболее рациональный метод и алгоритм решения задачи;
- Разработать алгоритм для решения прикладных задач;
- Написать программный код и отладить программу.

Выполнив работу, Вы будете:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- выполнять проверку, отладку кода программы;
- оформлять код программы в соответствии со стандартом кодирования;

Материальное обеспечение:

- Рабочее место преподавателя: персональный компьютер, проектор, рабочие места обучающихся, доска учебная, учебная мебель;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:

MS Windows 7 (подписка Imagine Premium)

MS Office 2007

7 Zip

MS Visual Studio (подписка Imagine Premium)

Visual Studio Code

Calculate Linux Desktop

Задание:

1. Формулировка задачи:

Написать метод $\min(x, y)$, находящий минимальное значение из двух чисел. С его помощью найти минимальное значение из четырех чисел a, b, c, d .

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

1. Формулировка задачи:

Написать метод $\max(x, y)$, находящий максимальное значение из двух чисел. С его помощью найти максимальное значение из четырех чисел a, b, c, d .

2. Создать внешний вид формы программы

3. Написать программные коды для созданной формы

Порядок выполнения работы

- Составить математическую модель задачи.
- Выбрать и обосновать наиболее рациональный метод решения задачи;
- Разработать алгоритм для решения задачи.

- Написать и отладить программу.

Ход работы:

- Выделить входные и выходные данные
- Построить блок-схему согласно типу алгоритмической конструкции
- Написать программный код на язык программирования и проверить работу кода с входными параметрами.

Форма предоставления результата

- Алгоритм программы.
- Код программы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.